

abc

N° 90

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Expert à domicile

Lire Pascal

Le nouvel Atari 130XE

Jeu : atterrissage sur EXL100

EDITIONS
ATLAS

**Page manquante
(publicité et colophon)**



Expert à domicile

Les systèmes experts sont des programmes hautement structurés susceptibles de tenir en grande partie le rôle de conseil et de diagnostic pour des professionnels.

Un spécialiste, qu'il s'appelle clinicien, géologue ou chimiste, est a priori une référence de crédibilité. Chacun a mis longtemps à apprendre sa spécialité; il a de l'expérience. Le problème, avec de tels experts humains, est qu'ils sont rares, ils ne sont pas toujours sûrs, ils sont chers et, finalement (!), ils meurent, emportant avec eux une grande partie de leur savoir. C'est pourquoi l'idée est venue aux informaticiens de mettre les connaissances sous forme de programmes, et de créer ainsi des bibliothèques de savoir.

Le concept des systèmes experts date des années soixante-dix, lorsque les chercheurs en intelligence artificielle ont abandonné, ou remis à plus tard, la réalisation de machines globalement intelligentes. Ils se sont alors tournés vers la recherche de solutions bien définies appartenant au monde réel.

Les systèmes experts ont donc constitué une des premières applications des recherches en intelligence artificielle. Il existe d'ores et déjà des systèmes plus performants que l'homme dans les domaines du diagnostic médical, de l'interprétation des spectrogrammes de masse, de la classification des maladies des récoltes, et dans d'autres domaines. Il est donc important de savoir comment ils fonctionnent.



Ce qu'est un système expert

Un système expert est obligatoirement constitué autour d'une importante base de connaissances relatives à un problème donné. Ces connaissances sont en général organisées sous la forme d'un ensemble de règles qui permettent de partir d'hypothèses et d'aboutir à des conclusions. Cette approche des systèmes informatiques, fondée sur la notion de connaissances, représente un changement aux conséquences révolutionnaires. Elle vient se substituer à la formule habituelle « données + algorithme = programme ». La nouvelle architecture est centrée sur une base de connaissances et un « moteur d'inférences » : « connaissance + inférence = système expert ». Cette formule est similaire, mais fondamentalement différente quant au fond.

Qu'est-ce qu'un système expert? La liste suivante de caractéristiques en donne une définition :

- Un système expert est limité à un domaine relativement étroit de connaissances.
- Il doit pouvoir raisonner avec des données incertaines et des règles non fiables.
- Il doit pouvoir exposer de manière complète le cheminement de son raisonnement.

- Les faits et le mécanisme d'inférence sont deux aspects distincts : les connaissances ne sont pas matériellement codées dans les procédures déductives.

- Il est conçu pour se développer.
- Il est par définition fondé sur des règles.
- Il donne, en sortie, des conseils et non des tables de chiffres ou des graphes.

Le mot clé est « connaissance ». Manifestement, l'objectif d'un système intelligent est de réduire les recherches se faisant au hasard ou de manière aléatoire.

A cette fin, le système informatique tire profit pratiquement du même atout que le spécialiste, à savoir des connaissances organisées. Connaissances factuelles, connaissances de règles d'inférences et de stratégies de résolution. Un système expert complet comporte quatre éléments :

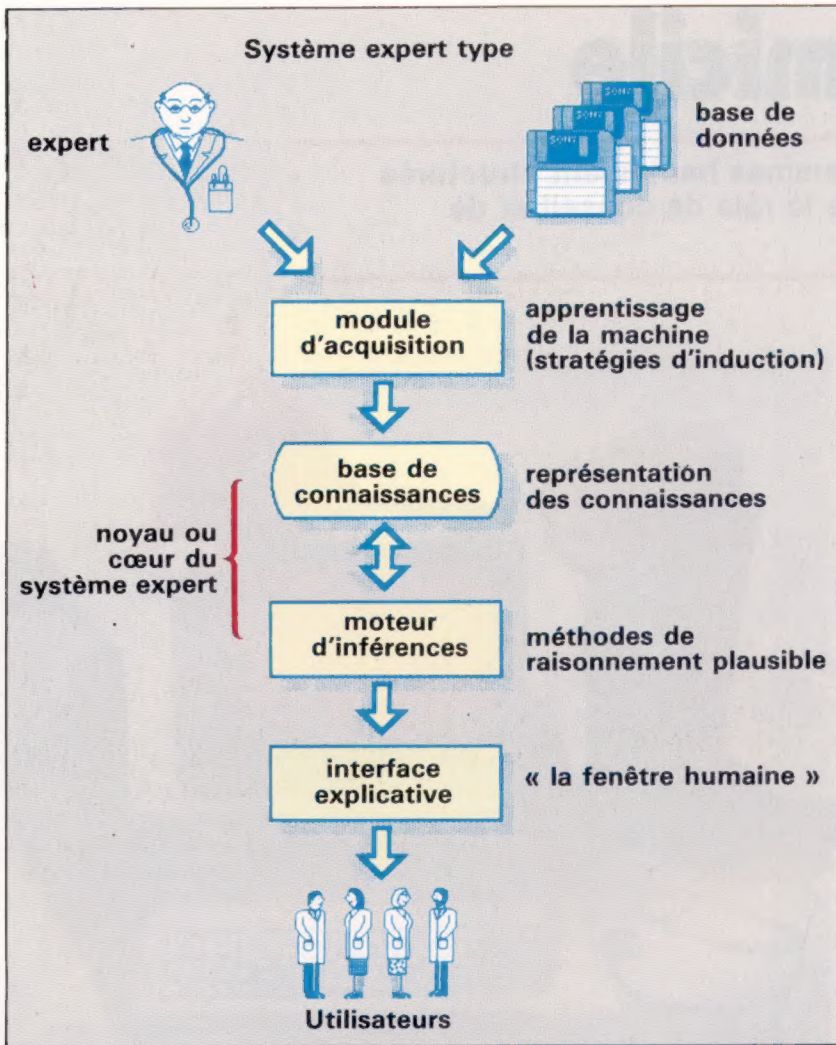
1. La base de connaissances.
2. Le moteur d'inférences.
3. Le module d'acquisition de connaissances.
4. L'interface explicative.

Les quatre modules sont essentiels. Un système fondé sur la connaissance pourra sans dommage manquer d'un de ces éléments, mais certainement pas un système expert.

Conseil de spécialistes

Les systèmes susceptibles d'acquiescer des connaissances auprès de spécialistes et de les utiliser pour donner ensuite des conseils ou établir des diagnostics sont de plus en plus répandus et concernent de nombreuses disciplines, depuis la médecine jusqu'à l'agriculture et l'architecture. Ils sont conçus pour être pertinents dans un domaine bien déterminé et jouent le rôle de consultants toujours disponibles.

(Cl. Marcus Wilson-Smith.)



Synopsis du système

Un système expert comprend plusieurs modules pour faire passer les connaissances du spécialiste à l'utilisateur final. Celles-ci doivent d'abord être acquises par le système et incorporées dans une base de connaissances. Pour donner des conseils ou effectuer un diagnostic, le système doit pouvoir tirer des conclusions à partir de la base (inférences). En dernier lieu, l'interface explicative permet à l'utilisateur de communiquer avec le système pour le consulter. (Cl. Ian McKinnell sur Macintosh.)

La base de connaissances

Les deux composants essentiels d'un système expert sont la base de connaissances et le moteur d'inférences. Le premier stocke des informations propres au sujet couvert. Les informations contenues dans une base de données ne constituent pas un ensemble passif d'enregistrements et de rubriques comme dans une base de données conventionnelle. Les informations consistent, pour un système expert, en représentations symboliques de règles d'appréciation et de jugement sous une forme qui permet au moteur d'inférences d'effectuer des déductions logiques.

La plupart des éléments d'une base de connaissances sont non mathématiques. Les deux principales difficultés de leur développement sont la représentation des connaissances et leur acquisition. Le premier problème concerne la manière de coder les informations pour que l'ordinateur puisse les utiliser. Il faut en général faire figurer les termes lexicaux propres au domaine de connaissances concerné (le jargon technique), les relations structurelles (les rapports mutuels entre éléments constitutifs) et les relations causales (les relations de causes à effets).

La tâche de l'ingénieur en système expert est de choisir le moyen approprié pour sauvegarder

ces informations de manière symbolique. Quatre méthodes principales se sont développées :

- Des règles du genre SI...ALORS. La condition spécifie une disposition, la conclusion pouvant être une action ou une assertion.
- Réseaux sémantiques. Ils représentent les relations entre les objets (par exemple, la baleine est un mammifère) par des liaisons entre nœuds-logiques.
- Structures. Ce sont des structures généralisées d'enregistrements qui peuvent prendre une valeur par défaut et comporter des actions codées comme valeurs de zone ou de case.
- Clauses d'appel. Il s'agit de logique des prédicats propre à PROLOG, lui permettant d'effectuer des inférences.

Les premiers systèmes experts utilisaient quasi exclusivement le formalisme fondé sur le principe des règles. Pour prendre une des règles du système de diagnostic des infections du sang, on voit une structure type SI...ALORS :

SI :

1. L'infection à soigner est une méningite, et
2. Le type de l'infection est cryptogamique, et
3. Aucun organisme n'est apparu sur l'élément de culture, et
4. Le patient n'est pas en danger, et
5. Le patient a séjourné dans une région où la coccidiose est endémique, et
6. Le patient est d'ethnie noire ou d'Asie ou d'Inde, et
7. L'antigène cryptocoque n'était pas positif

ALORS

Il est manifeste que le cryptocoque ne figure pas parmi les organismes responsables de l'infection.

Nous pouvons voir à partir de cet exemple qu'un système expert utilise le jargon du domaine où il intervient, ici la médecine. La construction syntaxique SI...ALORS utilisée est un ensemble d'énoncés pouvant être soit VRAI, soit FAUX. Ils peuvent donc être reliés par des opérateurs booléens, tels que ET, pour faciliter leur traitement par l'ordinateur. Pour faire apparaître l'information nécessaire à l'établissement d'un diagnostic, le système entre en dialogue avec l'utilisateur. Manifestement, ici tout du moins, l'utilisateur dispose d'un certain niveau de connaissances en la matière, afin que les questions très techniques de l'ordinateur puissent être comprises.

Le moteur à inférences

Les mécanismes d'inférence relèvent de méthodes de recherches et de raisonnement qui permettent au système de trouver des solutions. Elles permettent également, si nécessaire, de pouvoir fournir des justifications aux réponses données. Il existe deux stratégies d'ensembles : l'enchaînement vers l'avant, et l'enchaînement vers l'arrière.

Le premier type suppose de progresser à partir de ce que l'on sait (les symptômes), jusqu'aux conclusions (le diagnostic). Dans un système fondé sur un ensemble de règles, cela suppose d'accorder simplement les conditions SI avec les faits, éventuellement selon un ordre prédéterminé. L'enchaînement vers l'avant peut être faci-



lement informatisé et convient aux cas où l'ensemble des données doit être réuni de toutes manières. Comme exemple, on peut citer le cas où les données sont générées automatiquement par un instrument, et où des formulaires sont remplis.

L'enchaînement arrière progresse depuis une hypothèse jusqu'à une preuve. Il peut être programmé de manière réursive. Avec des systèmes de type consultation, cela donne lieu à un dialogue homme-machine beaucoup plus naturel. Le problème du choix de l'hypothèse, selon la situation, n'est toujours pas entièrement résolu. Aussi la plupart des systèmes mettent en œuvre un mélange des deux types d'enchaînements, vers l'avant et vers l'arrière.

Le module d'acquisition

Il est bien connu que des spécialistes ne peuvent pas toujours dire comment ils sont parvenus à leurs conclusions. Cela n'est pas nécessairement dû au fait qu'ils protègent les secrets de leur découverte, mais plutôt à ce qu'une grande partie de leurs raisonnements est inconsciente et d'ordre intuitif. C'est pourquoi l'acquisition des connaissances est considérée comme le point le plus délicat, le goulot d'étranglement du développement des systèmes experts. Les spécialistes restent néanmoins les meilleurs critiques. Ils peuvent considérer un cas type et dire quelle décision aurait dû être prise, et contester ainsi les solutions proposées par l'ordinateur. C'est pour tenir compte de cet enrichissement possible du système expert par l'homme, que des logiciels ont été spécialement développés récemment pour permettre aux systèmes d'induire leurs propres connaissances à partir d'exemples préclassés. Cela facilite effectivement l'acquisition des connaissances en surmontant à la fois la difficulté d'obtenir des informations de la part des spécialistes, et celle de les coder.

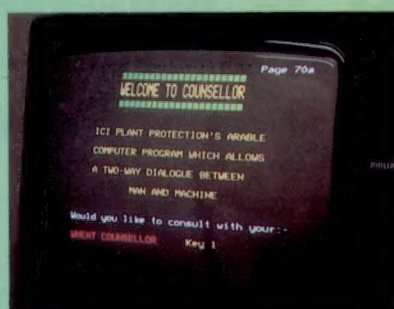
Même si les systèmes ne peuvent faire à eux seuls la totalité de la tâche (par exemple, d'une base de données à un jeu de règles de décision), il est déjà très précieux qu'ils puissent affiner leurs propres connaissances pendant une période « d'apprentissage », ou en cours d'utilisation en apprenant à partir de leurs propres erreurs. Nous verrons quelques machines d'acquisition de connaissances.

L'interface explicative

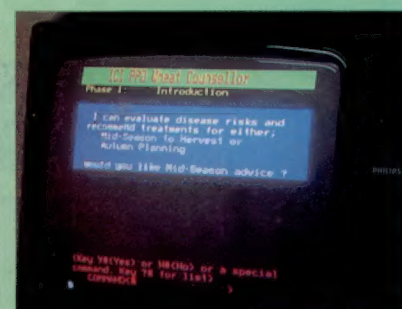
Un des points forts des systèmes experts classiques est leur capacité à pouvoir justifier leurs conclusions selon la demande. Il est en effet tout à fait essentiel que des décisions, dont peut dépendre la vie d'un patient ou, plus simplement, un gain ou une perte, puissent être vérifiées par l'homme.

Voici un exemple d'une réponse à la question POURQUOI?, posée ici par un praticien (qui n'accepte pas aveuglément les décisions du système expert). Le système commence la conversation en demandant des données :

Soyez bien conseillé...



L'introduction au programme permet de choisir entre consultation d'automne et consultation de printemps.



Comme en France à l'aide du Minitel et du service Télétel, les Britanniques peuvent accéder au programme Conseils en céréales à l'aide du service Prestel.



Le système pose alors des questions pour fonder son opinion.



Après avoir interprété les réponses du fermier, Conseils en céréales donne ses conseils sous forme de dosage de divers traitements chimiques.

la méningite est-elle une infection acquise en milieu hospitalier?
POURQUOI?

[2.0] la réponse devrait permettre de déterminer les micro-organismes incriminés (et autres que ceux identifiés sur cultures ou traces). Il a en outre été déterminé que :

[2.1] l'infection à soigner est une méningite, et

[2.2] l'étude par colorants de la culture n'a pas permis de mettre en évidence de micro-organismes, et

[2.3] l'infection est d'ordre bactériel, et

[2.4] la neurochirurgie est un des éléments significatifs de l'histoire médicale du patient, et

[2.5] l'âge du patient est supérieur à 10 ans. Aussi, si :

[2.6] l'infection du patient a été contractée lors de son hospitalisation, alors les micro-organismes (autres que ceux vus sur les marquages et cultures), probablement responsables de l'infection, sont e.coli (0,75), staphylocoque-coagpos (0,3), pseudomonas-aeruginosa (0,3), klebsiellapneumonia (0,5)

- selon (RÈGLE 545)

Les explications données sont rendues possibles en retraçant le cheminement du programme et en le restituant en termes médicaux. Elles sont facilement obtenues sur les systèmes fondés sur des règles, et à un faible coût (en indiquant les règles utilisées et celles qui précèdent). C'est du reste un argument en faveur du codage des connaissances selon le concept des règles. Vous remarquez que les conclusions comportent des coefficients de pondération. Il ne s'agit pas à proprement parler de probabilités, mais de simples facteurs qui permettent au système de donner de la cohérence à son univers d'incertitude. C'est une liste ordonnée de diagnostics probables.

Le bon grain

« Conseils en céréales » est un système expert développé par ICI en Grande-Bretagne et proposé gratuitement aux agriculteurs sur Prestel. Le système est destiné à conseiller les fermiers dans la lutte contre les maladies des céréales. Il donne les doses à appliquer, ainsi qu'une estimation des pertes probables pour les récoltes. Pour réunir les informations dont il a besoin en vue d'établir son traitement, le système pose une suite de questions de façon à cerner le problème. Son avis est enfin donné sous la forme d'une liste de traitements possibles accompagnés des doses appropriées. (Cl. ICI Plant Protection UK Division.)



Vue d'ensemble

Nous poursuivons notre étude de CP/M en nous intéressant aux fichiers sur lesquels les commandes travaillent, et aux façons de les manipuler.

Nous avons déjà parlé des extensions utilisées par CP/M, mais il est bon de les examiner plus en détail. Comme nous l'avons vu, un fichier CP/M se compose pour l'essentiel d'un nom primaire, comportant jusqu'à huit lettres, suivi d'un point et de trois lettres qui constituent une extension fichier (encore que cela n'ait rien d'obligatoire). Théoriquement, il est possible d'ajouter au fichier n'importe laquelle de ces extensions; mais certaines sont réservées à un usage précis. C'est ainsi que `COM` ne peut être utilisé que pour les fichiers de commande, et signale au système d'exploitation que le fichier correspondant doit être ajouté à la liste des commandes transitoires qu'il est possible de mettre en œuvre.

De la même façon, les programmes `BASIC` tournant sous CP/M doivent être suivis de l'extension `.BAS`. La plupart du temps, vous n'aurez pas à le préciser, car de nombreuses versions de `BASIC` utilisées en conjonction avec le système s'en chargent automatiquement. Le programme sera stocké sous forme de fichier source (texte). La différence avec un fichier texte à proprement parler est que lors d'un chargement ultérieur (préalable à l'exécution), CP/M se chargera de le « tokeniser » plutôt que de le lister comme il le ferait pour un fichier séquentiel ASCII. Il faut noter cependant que de nombreux ordinateurs compilent leurs programmes `BASIC` avant exécution. Les fichiers de ce type seront suivis de l'extension `.INT`, qui les désigne comme étant des fichiers `INT`ermédiaires abritant un programme objet.

Les programmes en langage machine peuvent eux aussi être suivis de diverses extensions, en fonction de leur statut. Lorsqu'on écrit un programme en assembleur, le listage source comportera l'extension `.ASM`. Faute de quoi, l'assembleur intégré ne cherchera pas à manipuler le tout et se bornera à afficher un message d'erreur.

Pour être possible, l'assemblage du programme doit être précédé par un choix entre deux extensions, `HEX`

ou `.PRN`. La première permettra la création, en fin de parcours, d'un fichier objet en `HEX`adécimal. La seconde enjoint à l'assembleur de produire de surcroît un listage imprimé de l'assemblage, accompagné d'un exemplaire du listage source, du programme objet, des adresses de chacun des codes opératoires, et de la liste des affectations, le tout avec le relevé des erreurs. On voit donc que `.PRN` est un outil essentiel pour le débogage.

Nous nous intéresserons pour terminer aux extensions qui accompagnent les fichiers texte. De nombreuses versions de CP/M disposent de capacités d'édition élémentaires (qui sont appelées par exécution du fichier de commande `ED`) intégrées au système d'exploitation proprement dit; mais d'autres vous contraindront à faire usage d'un logiciel de traitement de texte spécialisé, qui devra donc, lui aussi, tourner sous CP/M. Dans les deux cas, cependant, les extensions seront les mêmes.

A l'exception de quelques-unes déjà mentionnées, toute extension peut se composer de n'importe quel groupe de trois lettres. En fait, il n'est pas utile d'ajouter quoi que ce soit. Un fichier texte sauvegardé sans être suivi d'une extension se voit souvent ajouter quelque chose par le système. Le programme utilitaire `TEXTED` lui adjointra `.TXT` et `WordStar $$$`.

Copies de sauvegarde

Si vous jetez un coup d'œil au répertoire après la sauvegarde d'un fichier texte, vous constaterez qu'en fait *deux* fichiers ont été créés; le premier suivi de l'extension que vous avez définie, le second suivi de `.BAK`. C'est là une procédure de sécurité propre à CP/M. Il est bien sûr vital que des documents importants ne soient pas supprimés ou endommagés par accident; la méthode permet d'éliminer une telle possibilité puisque vous avez désormais deux copies de votre texte, un fichier normal et un fichier de copie, suivi de l'extension `.BAK`.

En cas de destruction accidentelle d'un fichier (par l'emploi inconsidéré de la commande `ERA`), vous pourrez toujours vous reporter à la copie, ce qui vous évite de tout recommencer à zéro. Un fichier édité, puis sauvegardé de nouveau, se voit accompagné d'une nouvelle version de `.BAK`. C'est donc une bonne idée de procéder de cette façon à intervalles réguliers, de sorte que la copie soit toujours à jour.

Il n'est pas indispensable d'ajouter des extensions, mais, une fois mises en œuvre, elles facilitent l'organisation de vos fichiers. Non seulement elles fournissent des informations sur leur structure et sur les entêtes, mais de plus elles favorisent la manipulation des groupes de fichiers. L'emploi de caractères de remplacement (*wildcard*) est de ce point de vue très utile quand on cherche à classer les fichiers selon des critères un peu approximatifs.

Supposons qu'un cadre supérieur ait écrit un certain nombre de rapports au cours du mois de juillet. Il peut estimer par la suite que plusieurs d'entre eux ne sont

Les caractères de contrôle de CP/M

Les caractères de contrôle permettent d'exécuter de nombreuses fonctions de CP/M. Certains sont aujourd'hui dépassés, et donc rarement mis en œuvre, mais ils sont conservés par le système d'exploitation afin de préserver autant que possible la compatibilité entre les différents ordinateurs.

CTRL C	Recharge en mémoire le disque système CP/M.
CTRL C	Initialise une nouvelle disquette.
CTRL M	Met un terme à la commande PIP, et repasse le contrôle au programme principal.
CTRL Z	Repasse le contrôle à ED après une opération d'insertion.
CTRL P	Envoie du texte à l'imprimante. Recommencer met un terme à l'opération.
CTRL U	Supprime la ligne en cours.
CTRL X	Supprime la ligne en cours et renvoie le curseur au début.
CTRL M	Exécute la ligne de commande en cours (remplace RETURN).
CTRL H	Retour en arrière/Suppression.
CTRL E	Permet l'insertion d'une ligne de commande sans exécution de celle-ci.
CTRL R	Répétition de la ligne de commande.



plus à jour, et qu'il vaut mieux les supprimer. Pour cela il pourrait, bien entendu, parcourir tout le répertoire et annuler chaque fichier individuellement à l'aide de la commande ERA; mais s'il a affaire à un grand nombre de textes, l'opération sera longue et fastidieuse.

S'il a pris soin de faire suivre ses fichiers de la même extension — disons .JUL — une seule commande permettra d'obtenir le même résultat : ERA D:* .JUL (D est optionnel et correspond au lecteur de disquette mis en œuvre). Placer un astérisque avant le point amène CP/M à négliger le nom de fichier proprement dit et à supprimer tous ceux qui seront suivis de l'extension .JUL. Il est aussi possible de le placer à droite du point, et même des deux côtés, comme dans ERA*.*. Mais, dans ce cas, il faudra prendre garde : tous les fichiers de la disquette seront effacés...

Suppressions sélectives

Revenons à notre cadre supérieur et partons du principe qu'il a donné à tous ses fichiers de juillet la même extension .JUL. Maintenant, compliquons un peu les choses en imaginant qu'il y a parmi eux des documents très importants qu'il ne doit à aucun prix effacer. Il sera hors de question, dans ce cas, de faire usage du préfixe *, qui provoquerait la destruction de tous les fichiers, quelle que soit leur valeur. Toutefois, notre responsable a pris soin de distinguer ses rapports des autres textes en les intitulant MEM1.JUL, MEM2.JUL, et ainsi de suite. Il sera possible de les faire disparaître à l'aide d'une seule commande, sans toucher aux autres. Il suffira de faire ERA D : MEM?.JUL. Le système d'exploitation reçoit ainsi l'ordre de supprimer tous les fichiers commençant par MEM et se terminant par le suffixe .JUL. Le ? placé en quatrième position indique que le caractère correspondant n'a pas d'importance particulière et ne doit pas être pris en compte.

Ce processus peut s'adapter aux onze positions possibles dans un nom de fichier. Notre cadre pourrait par exemple effacer d'un seul coup ses rapports de JUIN et de JUILLET, en spécifiant simplement MEM?.JUL. Il est même possible de faire apparaître simultanément * et ? dans le titre spécifié : ERA D:??????. permettrait par exemple de détruire tous les fichiers dont le nom comporte un Q en cinquième position.

Des caractères de remplacement tels que ceux-là peuvent de surcroît être utilisés avec d'autres commandes telles que PIP, STAT et REN. Pour faire passer un certain nombre de fichiers d'une disquette à une autre, on peut employer PIP B:=A*.*. Cela revient à donner à CP/M l'ordre de trouver sur la disquette A tous les fichiers correspondant au format spécifié (ici, tous), puis de les recopier sur la disquette B. Les opérations de copie sont de cette façon simplifiées au maximum.

Jusqu'à présent, nous nous sommes surtout intéressés aux commandes mises en œuvre pour contrôler CP/M. Mais le programme recourt fréquemment aux caractères de contrôle pour effectuer de nombreuses opérations. Le plus utile d'entre eux est sans doute la combinaison CTRL-C. L'appui de ces deux touches provoque un « redémarrage à chaud » du système, et entraîne un nouveau chargement en mémoire de CP/M. Cela n'est pas seulement utile quand le programme s'est planté mais aussi lorsqu'on met en place une nouvelle disquette dans l'un des lecteurs.

Il faut nous souvenir, en effet, que CP/M conserve en mémoire un exemplaire du « journal » de la disquette. Lorsque celle-ci est remplacée par une autre, il faut en informer le système, faute de quoi l'échange provoquera des erreurs. Un redémarrage à chaud de CP/M entraîne un nouveau chargement en mémoire du contenu de chacun des lecteurs, ce qui nous permettra de continuer les opérations. CTRL-C autorise par

ailleurs l'interruption de nombreux programmes, puis-que CP/M met automatiquement un terme au fonctionnement du système à chaque fois qu'il se remet à zéro.

Parmi les autres caractères de remplacement, beaucoup sont consacrés à l'édition de textes et ont été repris par plusieurs traitements de texte qui tournent sous CP/M. C'est ainsi que CTRL-H annule le dernier caractère, tandis que CTRL-U et CTRL-X effacent une ligne entière. Si une imprimante est raccordée à votre ordinateur, CTRL-P permet de lui transmettre du texte; l'opération peut être interrompue en tapant de nouveau cette combinaison.

Il est à noter que certains caractères de contrôle utilisés par CP/M ont été rendus inutiles par les progrès technologiques, au niveau du matériel comme à celui du logiciel. Pourquoi les avoir conservés? Il y a plusieurs réponses à cette question. Ils sont parfois redondants — c'est ainsi que la plupart des ordinateurs actuels comportent une touche DELETE. Ce n'était pas le cas autrefois (il y a quelques années à peine); les appareils de cette époque en étaient dépourvus et n'avaient pas non plus de touches curseur, aussi fallait-il intégrer des capacités de ce genre dans le système d'exploitation.

Reste aussi l'épineux problème de la compatibilité. La touche CTRL et les caractères alphanumériques ont, malgré les années, conservé la même valeur ASCII; mais ce n'est pas le cas des nouvelles touches, ce qui amène presque inévitablement au maintien des caractères de contrôle afin de sauvegarder la compatibilité entre systèmes. Enfin, un usager qui depuis plusieurs années travaille avec CP/M s'est peu à peu habitué à se servir des touches en question; en changer l'obligerait à tout reprendre depuis le début.

Nous venons de passer en revue, de façon très générale, les commandes et les fichiers tels que CP/M les met en œuvre. Reste à voir les fonctions que le programme met en œuvre dans son propre système d'exploitation. Dans l'article suivant, nous descendrons dans les profondeurs de CP/M pour voir comment il est structuré et comment il communique avec l'ordinateur et ses périphériques.

Marche à l'ombre

L'utilisation de caractères de remplacement est une des caractéristiques importantes de CP/M. En les introduisant en différentes positions, l'ordinateur ignorera effectivement les caractères correspondants. Cela signifie que par l'attribution précise de noms de fichier, l'utilisateur peut manipuler l'ensemble des listes des fichiers classées sous différents en-têtes. (Cl. Liz Dixon sur Macintosh.)

Caractères de remplacement

ERA DI?????5.BAK

DIA10/05.BAK
DIAGRM05.BAS
DIA03/05.BAK
DIA10/05.TXT
DIA01/04.BAK
DIA03/06.TXT
DIA03/06.BAK
DIA17/05.BAK
DISPLY05.BAS



Lire pascal

Vous êtes déjà un fervent adepte de PASCAL, langage structuré et efficace; ou bien vous n'avez pas encore eu le courage ou l'occasion de vous lancer à sa conquête... Alors, voici des livres pour vous.



Pascal : norme ISO et extensions

PASCAL est aujourd'hui parvenu à la force de l'âge, vivifié par un stage à l'Organisation internationale de normalisation, ce qui lui a apporté d'importantes améliorations, notamment pour renforcer l'indépendance modulaire, et de nombreuses précisions sur certains aspects du langage en vue d'accroître la fiabilité et la portabilité des programmes.

Cet exposé présente l'intégralité du langage PASCAL selon la norme internationale ISO. Il intègre, en outre, diverses extensions à cette norme, relatives en particulier à la compilation séparée, à l'organisation de fichiers non séquentiels, au traitement des chaînes de caractères, etc. Manuel de référence par son caractère exhaustif, ce livre est aussi le manuel de l'utilisateur grâce à une série de programmes illustrant chacun des concepts du langage.

Par Patrice Lignelet.
230 pages. Format 16,5 x 24 cm.
Masson.

Pascal : 76 programmes pour tous

Les soixante-seize programmes présentés dans cet ouvrage abordent certains problèmes que nous retrouvons dans les finances, les mathématiques, les statistiques, le budget familial, etc.

Les listages fournis sont prêts à être recopiés et exécutés immédiatement. Ils s'adressent aussi bien à des débutants qu'à des programmeurs expérimentés qui veulent connaître de nouvelles possibilités du traitement informatique.

Par Lon Poole.
250 pages. Format 21,5 x 28 cm.
McGraw-Hill



Initiation Pascal

Le PASCAL... mais c'est très simple. Ce livre vous en convaincra. Il vous aidera à maîtriser l'informatique et à mettre la puissance de l'ordinateur à votre service. Pour tout faire : gestion, commandes industrielles, enseignement assisté, contrôles ménagers, jeux, art, etc. Toutes les commandes et instructions sont expliquées à l'aide d'exemples de programmes, fournis systématiquement avec le résultat de leur action. Le tout est émaillé de commentaires qui facilitent leur compréhension et leur assimilation. Le présent ouvrage s'est appuyé particulièrement sur l'utilisation du PASCAL UCSD et les exercices ont été réalisés sur un micro-ordinateur Apple II.

Par J.-C. Guillemot.
220 pages. Format 21 x 29,5 cm.
S.E.C.F./Editions Radio.

Apple Pascal sur le bout des doigts

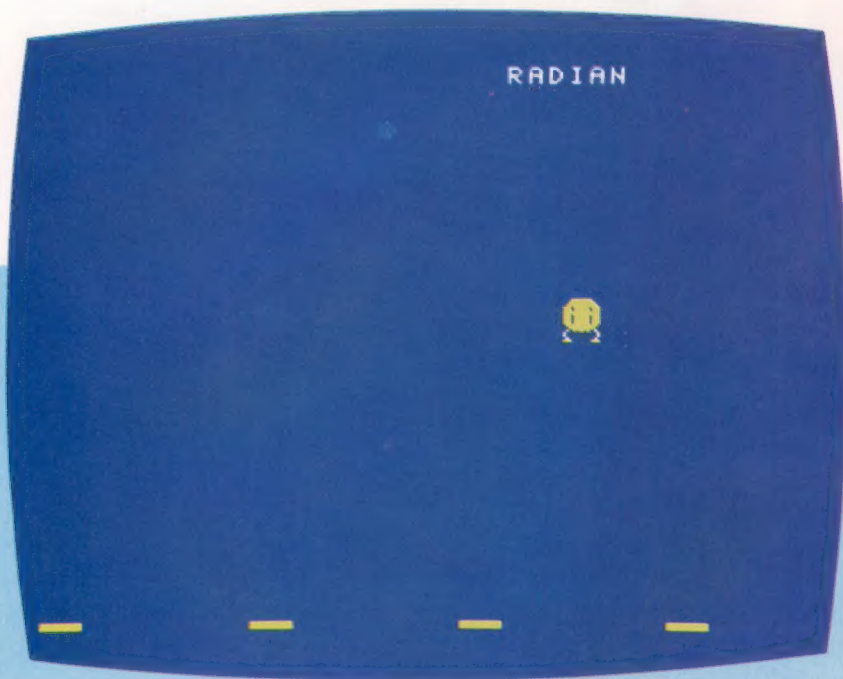
Bien que le principal objectif des auteurs soit de réussir à apprendre le langage PASCAL à un lecteur inexpérimenté, ils ne se sont pas limités à présenter un sous-ensemble banal du langage. Organisé comme une série de travaux pratiques soigneusement séquencés, cet ouvrage introduit aux blocs des procédures et des fonctions, à plusieurs types de données et aux principales structures de contrôle du langage. Dans la seconde moitié du livre, le lecteur apprendra à manipuler des structures de données qu'il aura définies lui-même : tableaux, ensembles, articles. L'ouvrage s'achève par un chapitre sur la récursivité.

Par Arthur Luehrmann et Herbert Peckham.
750 pages. Format 15 x 21 cm.
McGraw-Hill.

Atterrissage

Voici un jeu d'action, écrit par Pierre Monsaut, qui est toujours d'actualité. Mais n'oubliez pas que ce programme est destiné au micro-ordinateur EXL100.

Après un long voyage en apesanteur, poser une navette spatiale en douceur n'est pas chose aisée; mais, grâce à votre ordinateur, vous allez être en mesure de vous entraîner sans danger. Vous devez poser votre navette sur l'une des quatre aires prévues à cet effet. Vous pouvez vous diriger vers la droite et vers la gauche à l'aide des touches de contrôle du curseur.



```

100 REM *****
110 REM * ATERRISSAGE *
120 REM *****
130 GOSUB 590
140 FOR I=1 TO 100
150 NEXT I
160 IF DL<0 THEN DL=0
170 GOSUB 690
180 FOR Q=2 TO 20
190 FOR I=0 TO DL*10
200 NEXT I
210 CALL KEY1(D3,D4)
220 NH=NX
230 NX=NX+(D3=131)-(D3=129)
240 IF NX<2 THEN NX=2
250 IF NX>38 THEN NX=38
260 LOCATE (Q-1,NH)
270 PRINT C$;
280 LOCATE (Q,NH)
290 PRINT C$;
300 LOCATE (Q,NX)
310 PRINT N$;
320 LOCATE (Q-1,NX)
330 PRINT M$;
340 NEXT Q
350 IF INT((NX-3)/10)=(NX-3)/10 THEN DL=
DL-1:GOTO 140
360 CALL COLOR("1Yb")
370 LOCATE (Q,NX)

```

```

380 PRINT C$;
390 LOCATE (Q-1,NX)
400 PRINT C$;
410 LOCATE (Q+1,NX-1)
420 PRINT H$;
430 CALL COLOR("1Rb")
440 LOCATE (7,6)
450 PRINT "VOTRE NAVETTE S'EST ECRASEE";
460 LOCATE (12,12)
470 PRINT "SCORE :";S-1;
480 LOCATE (15,11)
490 PRINT "UNE AUTRE ?";
500 FOR I=1 TO 100
510 NEXT I
520 CALL KEY1(D3,D4)
530 IF D3<>255 THEN 520
540 CALL KEY1(D3,D4)
550 IF D3=255 THEN 540
560 IF D3<>78 THEN RUN
570 CLS
580 END
590 CLS ("Ybb")
600 GOSUB 800
610 B$=CHR$(32)
620 N$=CHR$(100)&CHR$(101)
630 C$=B$&B$
640 M$=CHR$(102)&CHR$(103)
650 A$=CHR$(104)&CHR$(104)
660 H$=CHR$(103)&CHR$(101)&CHR$(100)&CHR

```

```

$(102)
670 S=0
680 RETURN
690 CALL COLOR("1Yb")
700 FOR I=1 TO 20
710 NEXT I
720 CLS
730 FOR I=0 TO 3
740 LOCATE (22,3+I*10)
750 PRINT A$;
760 NEXT I
770 S=S+1
780 NX=INTRND(35)+2
790 RETURN
800 CALL CHAR(100,"0000001F3F7FEFEFFEF"
)
810 CALL CHAR(101,"000000E0F0BDCDFCDC"
)
820 CALL CHAR(102,"EFEFEF7F3F4B80B040E0"
)
830 CALL CHAR(103,"DCDCDFBF04B04040B1C"
)
840 CALL CHAR(104,"FFFFFF00000000000000"
)
850 DL=20
860 RETURN

```



Nouvel assaut

Avec son nouveau micro 130XE, Atari assure à la fois la compatibilité avec ses prédécesseurs XL et conserve ses excellentes fonctions graphiques et sonores. Une machine très compétitive.



Une nouvelle jeunesse
L'Atari 130XE est le premier produit de la société à être lancé depuis que Jack Tramiel en est devenu le président. Avec 128 K de RAM intégrés, à un prix qui se compare avantageusement à des ordinateurs offrant la moitié de la mémoire, Atari espère que cette machine lui donnera une place importante sur le marché de la micro-informatique. (Cl. Chris Stevens.)

Un des pionniers de l'informatique domestique, Atari, n'en a pas moins connu des années difficiles. En 1980, après avoir connu de lourdes pertes, la société a été reprise par l'ancien dirigeant de Commodore, Jack Tramiel. Sa politique, visant à proposer la technologie informatique aux meilleurs prix possibles, s'est vite traduite dans les faits, lorsque, dans les boutiques, les prix des micros 600XL et 800XL ont été diminués de façon importante pour la période de Noël.

Cette intervention ne fut pas suffisante pour inverser la tendance. Atari dut faire face à un problème commun à de nombreux constructeurs d'ordinateurs : les faibles chiffres de ventes entraînent une pénurie de logiciels, ce qui, par effet boomerang, amollit encore plus les ventes, et se traduit enfin par une raréfaction des fonds destinés à d'autres investissements.

Ironiquement, Jack Tramiel était le principal responsable de la position inconfortable d'Atari. La commercialisation efficace du Commodore qu'il avait réalisé a permis d'éclipser Atari comme concurrent. Depuis le début de 1985, la situation commence à changer. Commodore a dû faire face à une diminution des ventes et à l'échec de son Plus-4, tandis qu'Atari annonçait une

nouvelle gamme de produits. La première de ces nouvelles machines est l'Atari 130XE, un ordinateur construit autour du microprocesseur 6502C.

Le 130XE est presque identique à l'ordinateur Atari 8 bits, vendu sous diverses formes depuis le milieu des années quatre-vingt. La principale différence entre cette machine et les premiers ordinateurs de la société réside dans la nouvelle présentation et dans la capacité de mémoire vive : l'Atari 130XE offre 128 K de RAM.

Le boîtier de l'ordinateur est très différent de celui de ses prédécesseurs. L'extérieur gris pâle offre l'élégance que les consommateurs attendent de plus en plus d'un ordinateur moderne, avec des lignes arrondies et des touches moulées qui facilitent la frappe. Ces dernières sont de meilleure qualité que les versions précédentes et offrent une frappe plus confortable.

Comme les micros Atari fondés sur le 6502, le 130XE a cinq touches de fonctions préprogrammées. Cependant, contrairement aux versions précédentes, ces dernières ont été placées au-dessus du clavier principal et non du côté droit. Elles sont maintenant moulées dans le même plastique gris que le reste du boîtier et ont



la forme de parallélogrammes. Elles sont bien meilleures que celles des séries XL, qui étaient en métal et peu faciles à utiliser.

Connexions d'interface

Les interfaces, situées à l'arrière droit du nouvel Atari, offrent aussi quelques surprises. D'un côté, on retrouve les ports manche à balai habituels de type D à 9 broches utilisés en premier par ce constructeur et adoptés par presque tout le monde. A l'arrière de la machine, le port de commande série à 13 broches est utilisé pour connecter les unités périphériques Atari comme des unités cassette, des unités de disquettes et des imprimantes.

A la droite, à l'intérieur du boîtier, on aperçoit les interfaces de cartouches et d'extension. Le port cartouche permet à la machine d'exploiter la grande quantité de cartouches de jeux Atari-soft, comme Pacman et Galaxians, qui ont fait la force de la société au cours des dernières années.

Le port d'extension, cependant, est assez différent des normes précédentes. Les ordinateurs Atari avaient auparavant un connecteur plat à cinquante voies comme bus d'extension. La nouvelle machine possède un bus quatorze voies beaucoup plus petit servant de connecteur de cartouche. Les autres interfaces du 130XE sont les prises de moniteur composite, une fiche de jack HF pour les téléviseurs et l'alimentation standard Atari.

Pour proposer le 130XE au meilleur prix, la machine n'a pas de modulateur intégré lui permettant d'utiliser les téléviseurs de normes européennes, mais uniquement ceux qui répondent aux normes américaines. Il a donc été nécessaire de placer un petit boîtier (adaptateur) supplémentaire entre l'ordinateur et le téléviseur pour modifier le signal imagé.

L'esthétique et la compatibilité sont évidemment des atouts, mais Atari les a appliqués à la série XL sans véritable succès. Ce qui rend le 130XE différent des autres machines, et qui constitue un véritable argument de vente, est sa grande capacité de mémoire à faible prix. Un processeur 8 bits ne peut évidemment adresser que 64 K de RAM à la fois.

Pour adresser le double de RAM, l'ordinateur doit utiliser un processus dit « commutation de blocs ». A l'aide de cette technique, l'ordinateur peut utiliser une « fenêtre » de 64 K dans le total de 128. Bien que cette technique ne soit pas parfaite et que les instructions supplémentaires requises pour passer d'un bloc de RAM à un autre ralentissent l'exécution et l'extraction, il est ainsi possible pour un micro 8 bits d'adresser plus de mémoire.

En fait, la technique de commutation de blocs de mémoire est plus répandue que vous ne pourriez le croire. L'Atmos et le Commodore 64 disposent de plus de 64 K de mémoire et utilisent tous deux la commutation de blocs pour exploiter plus efficacement la mémoire disponible.

Port d'extension

La seule différence du 130XE par rapport aux ports périphériques des anciennes machines est la fente d'extension.

Prise HF

Le modulateur HF du 130XE est conforme aux normes américaines; le conducteur d'antenne est fourni avec un boîtier qui contient la logique permettant de répondre aux normes européennes.

Puces RAM

Les 128 K de RAM sont logés dans ces deux blocs de puces à 8 K.

Puce de commande mémoire

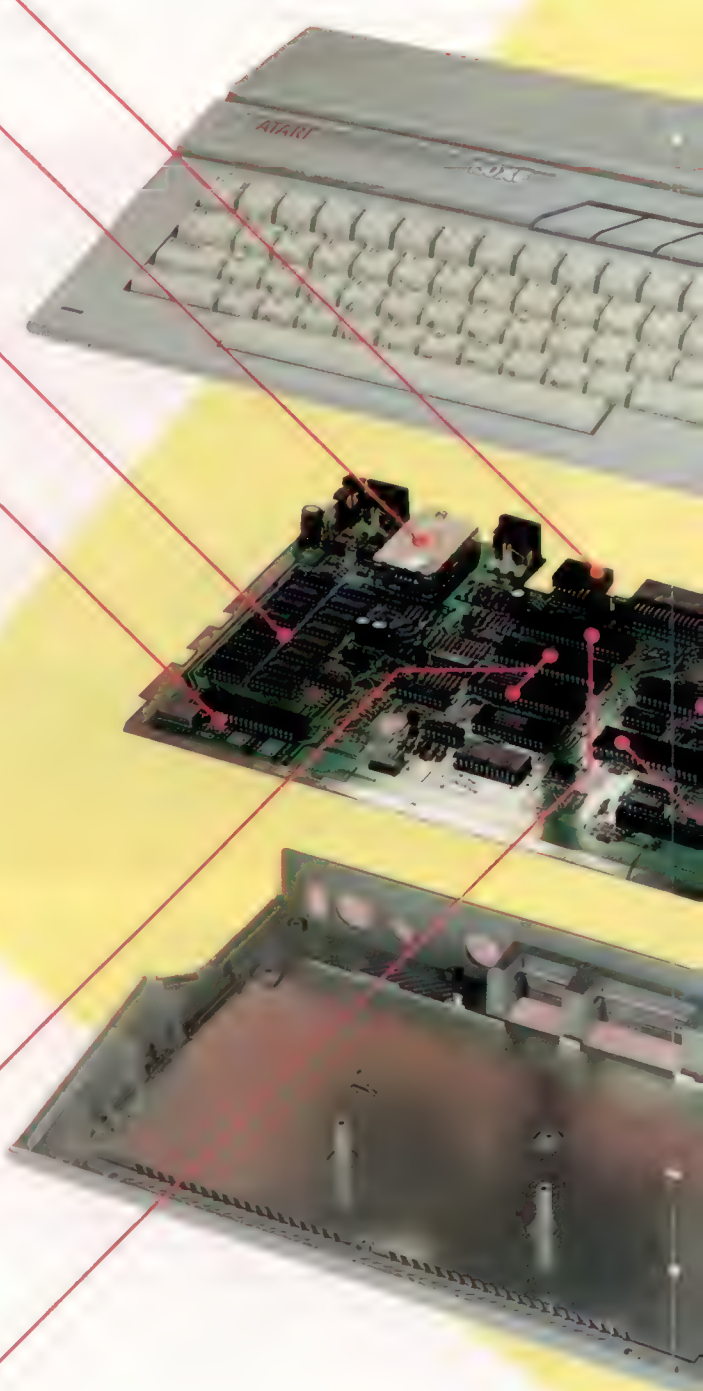
Les routines de gestion et de commutation mémoire sont contenues dans cette nouvelle puce qui a été nommée « Freddy ».

Puces graphiques

Les puces ANTIC et GTIA commandent les fonctions graphiques de l'écran de l'ordinateur.

UC

Comme toutes les machines Atari précédentes, le 130XE est construit autour du processeur 6502.





Port de cartouche
Le port de cartouche permet à l'ordinateur de tirer parti de la large variété de logiciels Atari.

Port périphérique
Des unités périphériques Atari, comme des unités de disquettes et des imprimantes, peuvent être reliées en chaîne à l'ordinateur au moyen de cette interface série.

Ports de manche à balai
L'ordinateur possède une paire de manches à balai qui naturellement répondent aux normes Atari.

Puce PIA
La commande d'entrée/sortie est assurée par une puce 6520.

Puce sonore
La puce POKEY est responsable du potentiel sonore à quatre octaves du 130XE.

Le prix du 130XE est assez étonnant. Bien que plus cher que le Spectrum ou que l'Electron, le 130XE est beaucoup moins cher que le Commodore 64. Afin de proposer un ordinateur de 128 K à ce prix tout en faisant des profits, Atari a dû réduire considérablement les coûts de production. Mais la nature de la machine elle-même a contribué à maintenir le prix assez bas : le 130XE est une machine « repensée », c'est-à-dire que les coûts de mise au point ont été réduits au maximum. Les principales économies ont été réalisées à l'intérieur de la machine.

Diminuer les coûts et automatiser

La zone de mémoire est composée de 16 puces RAM. Le coût de production de ces puces, qui ne font plus partie des produits de technologie de pointe, a baissé considérablement au cours des dernières années.

Une autre manière de réduire les coûts consiste à utiliser le minimum de composants. Bien que de nombreuses puces de la série XL réapparaissent dans le 130XE pour assurer la comptabilité, la carte de circuits imprimés est très bien conçue, et semble beaucoup moins encombrante que sur de nombreuses autres machines offrant moitié moins de capacité mémoire. Finalement, Atari a investi dans l'installation d'usines automatisées, et le 130XE est le premier produit à y être fabriqué. Tous les composants sont soudés à la machine.

Comme aucune puce sonore, graphique ou BASIC n'a été modifiée, l'ordinateur est, pour l'utilisateur, exactement identique aux modèles précédents, avec les excellentes performances techniques qui ont fait la réputation d'Atari. Le manuel est aussi amélioré. Les brochures qui accompagnaient les modèles précédents avaient tendance à être trop simplifiées, au point que l'on pouvait se demander si elles n'étaient pas destinées à des enfants. Le guide BASIC est amélioré et le constructeur a cette fois fourni de nombreuses précisions techniques dans l'annexe. Cependant, pour obtenir une explication exhaustive concernant le dialecte utilisé, l'utilisateur doit toujours acheter le manuel de référence du BASIC Atari.

La gamme de micros Atari devait absolument s'agrandir. Mais l'arrivée du 130XE soulève quand même des interrogations. Les 64 K supplémentaires offrent nettement plus de mémoire au programmeur, mais aucun programme n'en tire encore parti, même du côté des logiciels Atari compatibles déjà présents sur le marché. Normalement, on pourrait croire que le lancement d'une telle machine représente une première manœuvre visant à attaquer le marché de la petite entreprise. Cependant, la nouvelle administration Atari a nié une telle intention. Il s'agit peut-être de devancer le lancement du Commodore 128, une machine compatible avec le Commodore 64 qui a elle aussi une mémoire supplémentaire mais qui coûte beaucoup plus cher.

ATARI 130XE

PRIX

★ ★

DIMENSIONS

360 × 233 × 63 mm.

UC

6502C fonctionnant à 1,79 MHz.

MÉMOIRE

128 K de RAM, 24 K de ROM.

ÉCRAN

Affichage texte 40 × 24, 320 × 192 points (haute résolution) avec 256 couleurs disponibles.

INTERFACES

Port cartouche, jack T.V., prise moniteur composite, deux ports manche à balai, port entrée/sortie série.

LANGAGE DISPONIBLE

Atari BASIC, LOGO, FORTH, PILOT.

CLAVIER

62 touches, dont 5 touches de fonction préprogrammées.

DOCUMENTATION

Le manuel donne une explication complète du BASIC Atari, bien que le ton soit toujours un peu simpliste. On explique, dans l'annexe, les configurations d'interface et comment « passer » à l'autre bloc de 64 K de RAM.

FORCES

Le 130XE a l'avantage de pouvoir profiter des qualités Atari et possède en plus 64 K de mémoire additionnelle qui peuvent servir pour stocker et extraire rapidement des données.

FAIBLESSES

L'ordinateur ne résout pas le problème fondamental d'Atari : l'absence de logiciels « sérieux ». Le fait que le 130XE ne soit qu'une version améliorée d'une machine qui existe depuis de nombreuses années pourrait compromettre le succès espéré par le constructeur.

Recherches

Le déroulement d'un programme PROLOG ne suit pas le même schéma que celui des langages comme le BASIC ou le PASCAL. Arrêtons-nous sur la procédure arborescente de recherche de PROLOG.

Alors que les langages tels que le BASIC ou le PASCAL présentent un déroulement linéaire (le contrôle passant d'une instruction à l'autre dans un ordre strictement séquentiel même si le BASIC connaît les boucles et les interrogations GOTO) PROLOG va d'abord le plus loin possible dans les clauses du programme.

Pour mieux le comprendre, il faut penser aux programmes PROLOG comme à des arbres.

Le but (la proposition à démontrer) étant la souche. Tous les résultats intermédiaires sont autant d'embranchements. Il y a de nombreuses méthodes pour explorer un tel arbre. PROLOG considère la branche la plus à gauche et la suit aussi loin que possible. Au fur et à mesure qu'il essaie une branche, il marque son passage. Lorsqu'il ne peut plus avancer, il remonte d'un cran et prend le premier embranchement libre, le plus à gauche, pour poursuivre. C'est ainsi que PROLOG parcourt la totalité de l'arbre, si nécessaire.

Une méthode exhaustive de recherche verticale passerait en revue tous les cheminements possibles, mais ce serait long. PROLOG sait ménager sa peine.

Une clause PROLOG est, nous l'avons déjà vu, une règle disant que l'objectif final est vrai lorsque tous les sous-objectifs le sont :

```
objectif :- sous-objectif1, sous-objectif2,
           sous-objectif3... etc.
```

et ainsi de suite. Cela se comprendra mieux si nous l'écrivons de la sorte :

```
Si sous-objectif1 est vrai
  ET sous-objectif2 est vrai
  ET sous-objectif3 est vrai
  ET etc...
ALORS objectif est vrai.
```

Du fait que les sous-objectifs sont soumis à la relation ET, la totalité de la clause sera fausse si un seul des sous-objectifs ne peut être prouvé. PROLOG prend les sous-objectifs de gauche à droite. S'il ne peut en prouver un, il s'arrêtera en ce point.

La méthode consistant à remonter dans le programme en cas d'erreurs rend sans importance l'ordre d'écriture du programme. L'avantage est que l'ordre de déroulement du programme devient sans importance, alors qu'il est essentiel en BASIC.

Le programmeur peut donc se concentrer sur la structure logique du problème.

L'accent mis par PROLOG sur un énoncé déclaratif du problème ne signifie pas que vous ne pouvez pas considérer vos programmes comme ayant une conduite procédurière. La clause PROLOG :

```
martien(X) :- nb de membres (X,7), nb de
             têtes(X,2) peut programmer en (X,cobol)
```

peut se lire de manière déclarative : « X est un martien si X a sept membres, deux têtes, et peut programmer en COBOL. »

Une lecture procédurière serait : pour prouver que X est un martien, prouver d'abord qu'il a sept membres, deux têtes, et enfin qu'il peut programmer en COBOL.

PROLOG n'est pas aussi fortement « typé » que les autres langages. Il attache également moins d'importance au type de données des arguments de ses termes. Aussi le terme `pred-(Argument)` peut être utilisé plusieurs fois avec la variable `Argument` affectée à un entier, un atome (tel que `martien`, `vénusien`, `d24`, etc.), ou à une liste. PROLOG traite ses types de données de manières diverses, permettant, par exemple, la manipulation arithmétique de nombres.

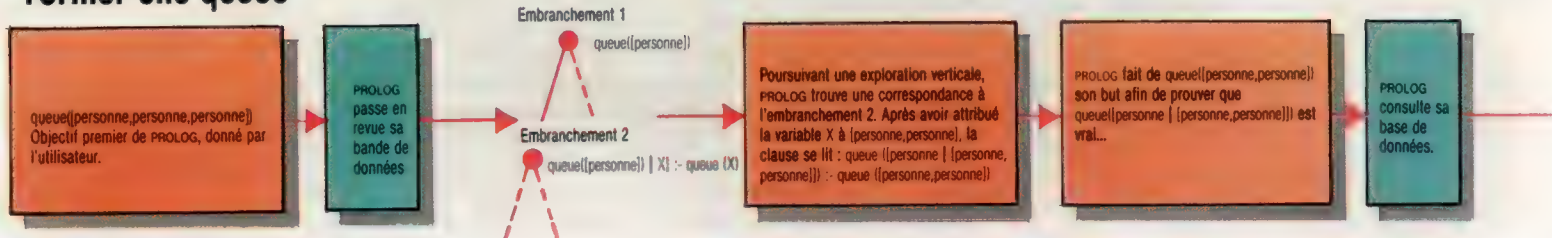
Si vous connaissez LISP ou LOGO, vous êtes déjà familier du type de données « liste », ainsi que des méthodes de traitement. Une liste PROLOG figure entre crochets, ses éléments entre virgules. Ainsi `[pomme, poire, banane]` est une liste de fruits ; `[a, f, e, g, r, x]` est une liste de lettres, et ainsi de suite.

Pour accéder à ce que contient une liste, PROLOG procède en retirant un élément à la fois à partir du premier. La notation `[Tête | Queue]` décrit une liste par l'élément `Tête`, le reste étant dans la liste `Queue`. Si on applique le symbole « | » à notre liste de fruits, nous obtenons `[pomme | [poire, banane]]`. Comme vous pouvez le constater, une liste peut comporter d'autres listes. Par exemple, si nous prenons la liste `[Z80]`, liste d'un seul élément, et si nous y faisons figurer le signe |, nous

Question temps

Ce déroulement de programme montre PROLOG répondant à une simple question. Vous remarquez que la variable `X` reçoit deux valeurs différentes lors de l'exécution du programme, les deux étant retenues.

Former une queue



, obtenons [Z80][1]. La liste de queue est [], ce qui représente la liste vide.

PROLOG autorise la récursion, qui est, en fait, le style normal d'un programme PROLOG. Avec PROLOG, nous pourrions écrire :

```
queue((personne)).
queue((personne | X)) :- queue(X).
```

Lorsque nous avons deux clauses ou plus, comme ici, qui ont la même tête, on a affaire à une procédure. La procédure de queue a une clause qui définit une queue comme étant une liste d'un élément, *personne*. Elle a ensuite une deuxième clause qui nous informe qu'une queue peut également être une liste avec l'élément *personne* en tête et une liste *X* à la fin. Nous voyons ensuite, d'après la partie droite de cette clause, que *X* doit être elle-même une queue.

Si nous donnons à PROLOG un objectif tel que :

```
queue((personne, personne, personne)).
```

lui demandant si la liste [*personne, personne, personne*] est une queue, il commence par rechercher dans sa base de données une clause correspondant à notre objectif.

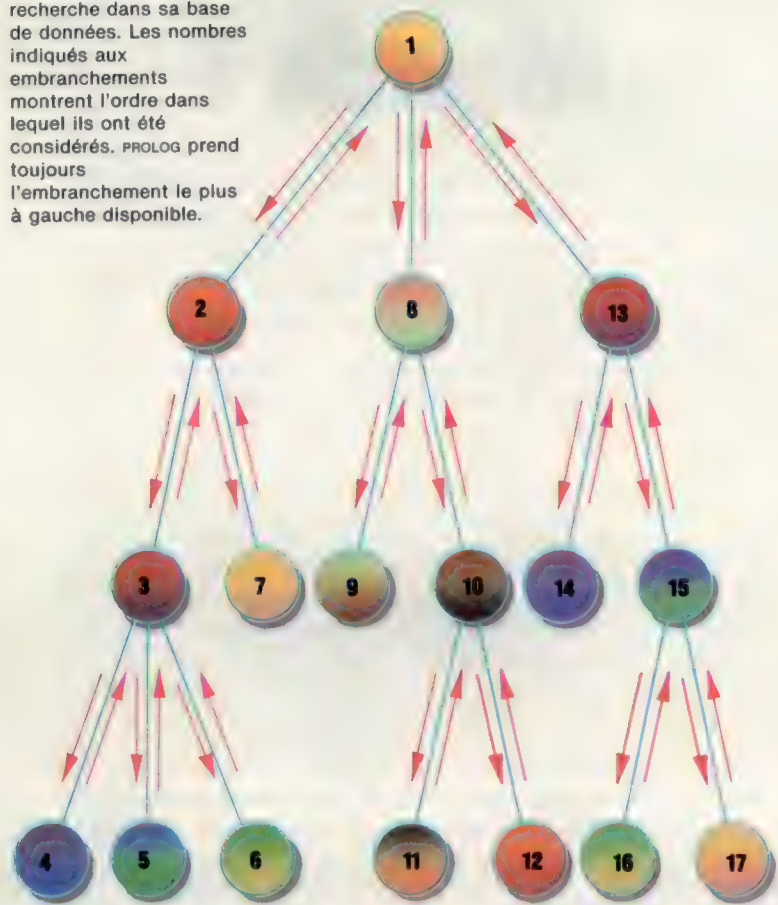
La première qu'il trouve est `queue((personne))`. Cela ne convient pas, du fait que les deux listes confrontées ne sont pas identiques. PROLOG passe alors à la clause suivante : `queue((personne | X)) :- queue(X)`. Cela ne correspond pas davantage et il donne alors des valeurs aux variables :

```
queue((personne | [personne, personne])) :- queue([personne, personne]).
```

Pour prouver que l'objectif de tête est vrai, il doit montrer que le sous-objectif est également vrai. Aussi PROLOG prend `queue((personne, personne))` comme objectif et entreprend de passer en revue toutes les clauses. A nouveau, `queue((personne))` ne correspond pas, mais la deuxième convient :

```
queue((personne | [personne])) :- queue([personne]).
```

Aller vers la gauche
PROLOG utilise une méthode verticale de recherche dans sa base de données. Les nombres indiqués aux embranchements montrent l'ordre dans lequel ils ont été considérés. PROLOG prend toujours l'embranchement le plus à gauche disponible.



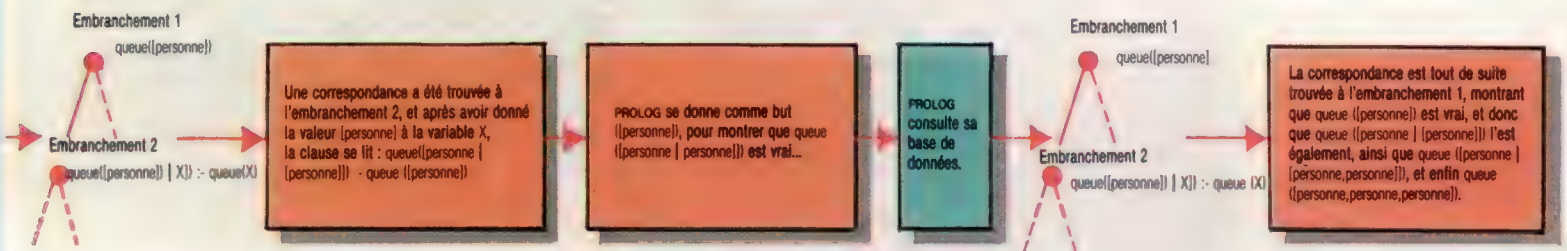
Un élément important à remarquer à ce niveau est que la variable *X*, affectée au premier passage à [*personne, personne*], l'est maintenant à [*personne*].

Cela est possible du fait que les variables sont locales à chaque appel séparé d'une clause. Ainsi, tous les appels de `((personne | X))` peuvent être considérés comme l'utilisation d'une seule et unique variable. La notion de variable globale n'existe pas avec PROLOG.

Ayant trouvé l'en-tête de notre objectif, le prouver passe par la preuve du sous-objectif à droite du

symbole `:-`, c'est-à-dire `queue((personne))`. Une autre recherche dans la base de données contenant les clauses est faite, et, cette fois-ci, PROLOG trouve « `queue((personne))`, qui se révèle vrai et établit la véracité de notre but précédent, qui lui-même prouve que le but d'origine est vrai. La procédure `queue` nous montre un certain nombre de points importants au sujet de PROLOG. Par exemple, que l'ordre des clauses peut être crucial (essayez de placer les deux clauses dans l'ordre inverse et voyez ce qui arrive lorsque vous voulez prouver un objectif).

Kevin Jones



Le Nouveau Monde I

Les derniers articles consacrés à notre jeu de simulation traiteront de variantes BASIC qui lui permettront de tourner sur Amstrad et Spectrum.

Le programme original est destiné au Commodore 64, mais son BASIC reste aussi minimal que possible. Pour le convertir à l'intention du Spectrum, deux grands problèmes se posent : le Spectrum n'accepte que des noms de variables réduits à une seule lettre, que ce soit dans les tableaux ou dans les boucles FOR...NEXT. Nous vous donnons ci-dessous une table de conversion pratique. Par ailleurs, le Spectrum manipule les chaînes de caractères de façon très particulière : il ne dispose pas de LEFT\$, RIGHT\$ ou MID\$, bien qu'il en propose toujours un équivalent. Reportez-vous aux modules déjà publiés, qui donnent à chaque fois les modifications à effectuer. Enfin, dès que vous verrez PRINT CHR\$(147), remplacez-le par CLS. Lorsque vous avez affaire à une ligne qui attend que vous appuyiez sur une touche, du type :

```
<no de ligne> GET I$: IF I$ = « » THEN <no de ligne>
```

remplacez-la par

```
<no de ligne> LET I$=INKEY$:IF I$ = « » THEN GOTO <no de ligne>
```

La prochaine fois, nous vous présenterons la deuxième partie du listage complet.

Table de conversion des variables pour Spectrum

Microsoft	Fonction	Spectrum
T\$(,)	Équipage (catégorie et force)	T(,)
WG(,)	Taux des salaires	W(,)
CC(,)	Décompte catégorie équipage	Y(,)
PA(,)	Provisions achetées	A(,)
PC(,)	Coût des provisions	C(,)
PN(,)	Besoins en provisions	N(,)
OC(,)	Coût des marchandises à échanger	O(,)
OA(,)	Quantités de marchandises à échanger	G(,)
HR(,)	Drapeaux demi-rations	H(,)
RR(,)	Drapeaux événements voyage	R(,)
AO(,)	Quantités marchandises échangées	E(,)
EQ(,)	Taux d'échange	Q(,)
V1(,)	Valeur au départ du port	B(,)
V2(,)	Valeur au retour	D(,)
S1	Réduit compteur force équipage	S
S3	Compteur impression lente	S
S4	Compteur bref délai	J
S5	Compteur long délai	S

```
1 REM *****
2 REM ** JEU DU **
3 REM ** NOUVEAU **
4 REM ** MONDE **
5 REM *****
6 :
```

La première section du programme initialise les variables et les tableaux qui seront mis en œuvre ultérieurement.

```
9 K$="" APPUYEZ SUR UNE TOUCHE POUR CONTINUER"
10 DIM TS(16,2): REM CATEGORIE ET FORCE EQUIPAGE
11 CN=0: REM NB. MEMBRE EQUIPAGE
12 MD=2000: REM ARGENT AU DEPART
13 DIMG(5):MG(1)=10:MG(2)=25:MG(3)=15:MG(4)=20:MG(5)=15: REM SALAIRES
14 MT=0: REM SALAIRES HEBDOMADAIRES
15 DM=16: REM EQUIPAGE MAXIMUM
16 DIMC(3):C(1)="MARIN":C(2)="MEDECIN":C(3)="CHARPENTIER"
17 D(4)="NAVIGATEUR":C(5)="CUISINIER"
18 DIMCC(5): REM DECOMPTE CATEGORIES EQUIPAGE
20 DIML(4):L(1)="KO":L(2)="KO":L(3)="KO":L(4)="KO"
1UB(4)="BARRIQUE"
21 DIMP(4):P(1)="LEO":P(2)="FRUITS":P(3)="VIRANDE"
1PA(4)="EAU"
22 DIMPA(4)
23 DIMPC(4):PC(1)=.5:PC(2)=1:PC(3)=2:PC(4)=.5
24 DIMPN(4):PN(1)=2:PN(2)=1:PN(3)=1:PN(4)=.5
30 DIMDA(6)
31 DIMDB(6)
32 DB(1)="FLACON DE MEDICAMENT":DB(2)="FUSIL":DB(3)="SAC DE SEL"
33 DB(4)="BALLON DE TIBBU":DB(5)="COUTEAU":DB(6)="BIJOU"
34 DIMDC(6)
35 DC(1)=1:DC(2)=5:DC(3)=.2
36 DC(4)=2:DC(5)=.5:DC(6)=1
40 JL=0: REM LONGUEUR DU VOYAGE
41 EA=0: REM SEMAINES EN PLUS
42 DIMRR(16)
43 REM INDICATEURS SI EVNTS ALEAT(N) DEJA SURVENUS
44 RC=0
45 REM DECOMPTE EVNTS ALEAT.
46 RR=13
47 DB="N": REM INDICATEUR BEAU TEMPS POUR DEFINITION FACTEUR MUTINERIE
48 AA="N":DB="N"
49 DIMH(6): REM INDICATEURS POUR SAVOIR SI EVNTS MAJEURS SURVENUS
60 DIMT(3):T(1)="PERLES":T(2)="STATUES":T(3)="EPICES"
61 DIMV1(3):V1(1)=2:V1(2)=2:V1(3)=1
62 DIMV2(3):V2(1)=2+INT(RND(1)*1)/2:V2(2)=2+INT(RND(1)*3)-1
63 V2(3)=2+INT(RND(1)*1)/2
64 DIMED(4,3)
65 ED(1,1)=0.5:ED(1,2)=0.5:ED(1,3)=1
66 ED(2,1)=5:ED(2,2)=5:ED(2,3)=10
67 ED(3,1)=5:ED(3,2)=5:ED(3,3)=6
68 ED(4,1)=2:ED(4,2)=2:ED(4,3)=4
69 DIMAD(3)
80 PRINTCHR$(147):B$="" VOYAGE DANS LE NOUVEAU MONDE""
00SUB9100:PRINT
81 00SUB9200
82 B$="VOUS COMMANDEZ UN NAVIRE"" 100SUB9100:PRINT
83 B$="PARTANT POUR LE NOUVEAU MONDE""100SUB9100:PRINT
84 B$="LE VOYAGE DURERA 8 SEMAINES""100SUB9100:PRINT
85 B$="PEUT-ETRE PLUS""100SUB9100:PRINT
86 B$="VOUS DEVEZ ENBAUCHER DES HOMMES. LES PAYER""100SUB9100:PRINT
87 B$="ACHETER PROVISIONS, EQUIPEMENTS ET MARCHANDISES""100SUB9100:PRINT
88 B$="VOUS DISPOSEZ DE 2 000 PIECES D'OR""100SUB9100:PRINT
89 B$="POUR VOS DEPENSES.""100SUB9100:PRINT100SUB9200
90 PRINTB$:" BONNE CHANCE ! "100SUB9100:00SUB9200:PRINT
91 B$="K":00SUB9100
94 DET1:IF I$="":THENSA
95 00SUB9200
```

Cette section du programme principal appelle les sous-programmes qui permettent au joueur de recruter un équipage, d'acheter des provisions pour le voyage, ainsi que les marchandises qui seront échangées avec les indigènes.

```
500 00SUB1000
350 00SUB2000
600 00SUB3000
610 PRINTCHR$(147)
615 B$="VOUS VOILA PRET A ENTREPRENDRE""100SUB9100
625 B$="VOTRE VOYAGE.""100SUB9100
630 00SUB9200
635 PRINTB$:"VOTRE EQUIPAGE COMPREND:"100SUB9100
9100
```




```

644 GOSUB 9200
645 FORT=1T05
650 IF CC(T)=8 THEN 670
655 PRINT CC(T):
660 PRINT C(T):
662 IF CC(T)=1 THEN PRINT " 100T0660
664 PRINT "S"
668 GOSUB 9200
670 NEXT
674 GOSUB 9200
675 PRINT S0="ET VOUS AVEZ COMME PROVISIONS:"*100S0
69100
692 GOSUB 9200
695 FORT=1T04
698 IF PA(T)=8 THEN 710
695 PRINT PA(T):U(T):"S DE":
700 PRINT P(T)
708 GOSUB 9200
710 NEXT
715 GOSUB 9200
720 PRINT S0="VOUS AVEZ AUGMENTE:"*100S0
725 GOSUB 9200
730 IF DR(1)=8 THEN 740
735 IF DR(1)=1 THEN S0="FLACON DE MEDICAMENT="*100T0735
734 S0="FLACONS DE MEDICAMENT="
735 PRINT DR(1):100S0
736 GOSUB 9200
740 IF DR(2)=8 THEN 750
743 IF DR(2)=1 THEN S0="FUSILS:"*100T0745
744 S0="FUSILS:"
745 PRINT DR(2):100S0
746 GOSUB 9200
750 IF DR(3)=8 THEN 760
753 IF DR(3)=1 THEN S0="SAC DE BEL:"*100T0755
754 S0="SACS DE BEL:"
755 PRINT DR(3):100S0
756 GOSUB 9200
760 IF DR(4)=8 THEN 770
763 IF DR(4)=1 THEN S0="BILLOT DE TISSU:"*100T0765
764 S0="BALLOTS DE TISSU:"
765 PRINT DR(4):100S0
766 GOSUB 9200
770 IF DR(5)=8 THEN 780
773 IF DR(5)=1 THEN S0="COUTEAU:"*100T0775
774 S0="COUTEAUX:"
775 PRINT DR(5):100S0
776 GOSUB 9200
780 IF DR(6)=8 THEN 790
783 IF DR(6)=1 THEN S0="BIJOU:"*100T0785
784 S0="BIJOUX:"
785 PRINT DR(6):100S0
786 GOSUB 9200
790 GOSUB 9200
792 PRINT "IL VOUS RESTE:"*INDI:"PIECES D'OR"
796 GOSUB 9200
797 S0="APPUYEZ SUR UNE TOUCHE POUR COMMENCER:"
798 GOSUB 9200
799 GET I0:IF I0="" THEN 799
800 WT=0:REM TOTAL SALAIRES MIS A ZERO
801 HS="N":REM INDICATEUR DES RATIONS
802 DIM HR(6):HR(1)=1:HR(2)=1:HR(3)=1:HR(4)=1

```

La boucle principale commence ici, et fait usage de la variable WK pour décompter le nombre de semaines passées en mer.

```

820 WK=1
823 GOSUB 4000:REM RAPPORT STATUT EQUIPAGE
828 GOSUB 4200:REM RAPPORT PROVISIONS
833 GOSUB 4300:REM RAPPORT AUTRES MARCHANDISES
840 GOSUB 9200:PRINT CHR(147)
842 PRINT:PRINT:PRINT
843 S0="LE VOYAGE DEVAIT PRENDRE:"*100S0
844 PRINT "ENCORE:"*INT(JL-WK+1):"SEMAINES DE PLUS"
845 GOSUB 9200
846 PRINT S0="K:"*100S0
847 GET I0:IF I0="" THEN 847
850 GOSUB 5000:REM VERIF DEPENSES SALAIRE
855 GOSUB 5100:REM DISTRIBUTION RATIONS
860 GOSUB 5200
861 REM GENERATION EVENEMENTS ALERTOIRES
870 GOSUB 6000:REM EVENEMENTS MAJEURS
875 IF HR(3)=.5 AND HR(1)<=.5 THEN PRINT CHR(147):GOSUB
8800
878 REM ALBATROS SI MANQUE DE VIRANDE
879 GOSUB 7000
890 GOSUB 5300:REM RAPPORT FIN DE SEMAINE
895 WK=WK+1:IF WK<=JL THEN 825
898 REM ARRIVEE NOUVEAU MONDE
899:GOSUB 10000
902:GOSUB 10070
904:GOSUB 10100
906:END

```

Le programme principal prend fin ici. Il n'y a plus ensuite que des sous-programmes appelés à partir de cette section.

```

1000 PRINT CHR(147):PRINT "PHASE 1 : EMBARCATION EQUIPAGE"
1010 PRINT "-----"
1012 PRINT
1015 GOSUB 10000

```

```

1020 PRINT:PRINT "CATEGORIES DISPONIBLES:"
1025 GOSUB 2000
1030 PRINT
1040 PRINT "TYPE DESCRIPTION SALAIRE MED00"
1050 PRINT
1060 PRINT " 1 MATELOT 10 PCS D'OR"
1070 PRINT " 2 MEDecin 25 PCS D'OR"
1080 PRINT " 3 CHARPENTIER 15 PCS D'OR"
1090 PRINT " 4 NAVIGATEUR 20 PCS D'OR"
1100 PRINT " 5 CUISINIER 15 PCS D'OR"
1105 GOSUB 9200
1110 PRINT:PRINT:PRINT
1120 S0="TAPEZ LE NUMERO CORRESPONDANT(1-5):"*100S0
1122 S0="ou 'F' SI VOUS AVEZ TERMINE:"*100S0
1125 INPUT I0
1126 CT=VAL(I0)
1129 IF LEFT$(I0,1)="" THEN PRINT:PRINT "FIN DE L'EMBARQUE."
1130 GOSUB 9200:GOTO 1130
1130 IF CT>5 AND CT<0 THEN 1150
1135 PRINT:PRINT
1140 PRINT I0:100S0="NON VALABLE:"*100S0
1142 GOSUB 9200
1145 S0="RECOMMENCEZ S.V.P."
1146 GOSUB 9200
1147 GOTO 1130
1150 PRINT:PRINT
1155 CN=CN+1:REM EQUIPAGE EMBARQUE
1156 TS=CN:1+AT:REM CATEGORIE
1157 TS=CN:2+100:REM FORCE AU DEPART
1158 WT=WT+(CT):REM TOTAL DES SALAIRES
1159 CC(CT)=CC(CT)+1:REM COMPTEUR CATEGORIES
1160 S0="L'EQUIPAGE JUSQU'A PRESENT:"
1170 FORT=1T05
1180 PRINT S0:CC(T):" "100(T)
1185 IF CC(T)>10000 THEN PRINT "S"*100T0189
1186 PRINT " "
1189 S0=" "
1190 NEXT
1195 PRINT:PRINT "TOTAL DES SALAIRES PAR SEMAINE:"*WT
1200 IF CN=1 THEN PRINT:PRINT S0="ENCORE UN HOMME:"*100S0
1201 GOTO 1295
1202 IF CN=1 THEN PRINT:PRINT S0="EQUIPAGE AU COMPLET:"*100S0
1203 GOTO 1310
1205 REM
1206 GOTO 1815
1310 PRINT:PRINT S0="K:"*100S0
1320 GET I0:IF I0="" THEN 1320
1325 GOSUB 9200
1330 PRINT CHR(147)
2000 PRINT CHR(147)
2010 S0=" PHASE 2 - LES PROVISIONS:"
2015 GOSUB 9200
2020 S0="-----"
2025 GOSUB 9200
2030 GOSUB 9200:PRINT
2040 PRINT "VOTRE EQUIPAGE COMPREND:"*INDI:"HOMMES."
2045 GOSUB 9200:GOTO 1770
2050 FORT=1T04
2055 PRINT
2060 PRINT "CHACUN AURA BESOIN:"
2070 PRINT "D'AU MOINS:"*INDI:"PIECES D'OR"
2075 IF PA(T)=1 THEN PRINT " "100T02085
2080 PRINT "S:"
2085 PRINT " DE:"*PA(T)
2090 PRINT "A:"*PC(T):"PCS D'OR PAR:"*U(T)
2095 PRINT "POUR CHACUNE SEMAINE DE VOYAGE."
2095 GOSUB 9200:PRINT:GOSUB 9200
2100 PRINT "COMBIEN DE:"*U(T):"S DE:"*PA(T)
2110 S0="VOULEZ-VOUS ACHETER?"*100S0
2120 PRINT
2130 INPUT I0
2140 PA(T)=VAL(I0):GOSUB 9200
2150 IF PA(T)>((CN+PA(T))-1) THEN 2260
2160 IF PA(T)=0 THEN PRINT "SI VOUS N'EN ACHETEZ PAS:"*100T02180
2180
2170 PRINT "SI VOUS N'ACHETEZ QUE:"*PA(T):U(T)
2175 IF PA(T)=1 THEN PRINT "DE:"*100T02180
2176 PRINT "S DE:"
2180 PRINT P(T):" "100S0
2190 PRINT "QUELQU'UN AURA PEUT-ETRE:"
2200 S0="FAIM"
2210 IF T=8 THEN S0="BOIF"
2220 PRINT S0:" "100S0
2230 S0="VOULEZ-VOUS RECOMMENCER (O/N)?"*100S0
2240 INPUT P0:PA=LEFT$(P0,1)
2242 IF P0<>"O" AND P0<>"N" THEN 2230
2245 IF P0="N" THEN 2400
2250 PA(T)=0:T=1:GOTO 2410
2260 IF PA(T)=PC(T) THEN 2270
2265 GOTO 2400
2270 S0="VOUS N'AVEZ PAS DE QUOI ACHETER:"*100S0
2280 PRINT PA(T)
2290 PRINT U(T):"S DE:"*PA(T):100S0
2300 S0="RECOMMENCEZ S.V.P.:"*100S0
2310 PA(T)=0:T=1:GOTO 2410
2400 MD=MD-(PA(T)-PC(T))
2410 PRINT:PRINT S0="LES PROVISIONS JUSQU'A PRESENT:"*100S0
2412 GOSUB 9200
2415 FORT=1T04
2420 PRINT PA(T):U(T):
2430 IF PA(T)=1 THEN PRINT " DE:"*100T02440
2435 PRINT "S DE:"
2440 PRINT P(T)
2450

```

```

2450 NEXT
2460 PRINT"IL VOUS RESTE "IHD1" PIECES D'OR "
3425 GOSUB9200
2490 NEXT
2500 GOSUB9201:PRINTI8="FIN DE L'APPROVISIONNEMENT="I00S
UB9100:GOSUB9200
2510 PRINT: S4="I00SUB9100:PRINT: GOSUB9200
2520 GETI0:IFI0=""THEN2520
2599 RETURN
3000 PRINTCHR(147): REM PHASE 3
3002 GOSUB9200
3005 PRINT" PHASE 3 - AUTRES MARCHANDISES "
3010 PRINT" ----- "
3020 GOSUB9200
3025 PRINT
3030 S4="D'AUTRES CHOSES PEUVENT ="I00SUB9100
3035 S4="ETRE INDISPENSABLES. AINSI ="I00SUB9100
3040 S4="POUR BOISSON LES MALADES OU COMMENCER ="I00SUB9100
3045 GOSUB9200
3050 S4="ET IL VOUS FAUDRA DES ARMES. ="I00SUB9100
3055 GOSUB9200:GOSUB9200
3060 FORT=1T04
3065 PRINT
3070 PRINT"CHIQUE "I06(T):
3075 S4="COUTE="I00SUB9100
3080 PRINTOC(T):
3081 PRINT" PIECES D'OR"
3085 IFOC(T)=1THENPRINT" I00T03090
3090 PRINT"PIECES D'OR"
3095 GOSUB9200
3095 S4="VOULEZ-VOUS EN ACHETER ? (O/N)="I00SUB9100
3110 INPUTP4:P4=LEFT(P4,1)
3115 IFP4<>"O"ANDP4<>"N"THEN3095
3120 IFP4="N"THEN3175
3125 GOSUB9200
3130 S4="COMBIEN EN VOULEZ-VOUS ?="I00SUB9100
3135 INPUTI0
3140 TT=VAL(I0)
3145 IFOC(T)*TT>MOTHS150
3147 DOTO3160
3150 S4="VOUS N'AVEZ PAS ASSEZ D'ARGENT !"="I00SUB9100
3155 GOSUB9200
3154 S4="RECOMMENCEZ S. V. P. ="I00SUB9100
3155 GOSUB9200:GOSUB9200
3160 MD=MO-(OC(T)*TT)
3165 OR(T)=TT
3170 GOSUB9200
3175 PRINT
3175 PRINT"IL RESTE "IHD1"PIECES D'OR. "IHD
3200 GOSUB9200:NEXT T
3205 GOSUB9200:PRINT:PRINT
3210 S4="FIN DE LA PHASE 3="I00SUB9100
3220 GOSUB9200:PRINT
3230 S4="I00SUB9100
3240 GETI0:IFI0=""THEN3240
3999 RETURN

```

Les sous-programmes qui suivent, appelés depuis la boucle principale, analysent la situation du navire et de l'équipage, et proposent au joueur un relevé hebdomadaire.

```

4000 REM RAPPORT SANTE EQUIPAGE
4010 PRINTCHR(147)
4020 S4=" JOURNAL DE BORD="I00SUB9100
4025 S4=" "I00SUB9100
4030 GOSUB9200
4035 PRINT"APRES"(MK-1)"SEMAINES"
4040 S4="L'EQUIPAGE EST ACTUELLEMENT ="I00SUB9100
4045 GOSUB9200:PRINT
4055 PRINT
4060 FORT=1T04
4070 IFB(T,1)=THENA110
4075 PRINTC(TB(T,1))" ("
4070 IFB(T,2)=999THENB="MORT !!!!!"="I00T04099
4080 IFB(T,2)>75THENB="EN PLEINE SANTE"="I00T04099
4085 IFB(T,2)>50THENB="EN BONNE SANTE"="I00T04099
4090 IFB(T,2)>25THENB="MALADE !"="I00T04099
4095 S4="TRES MALADE !"="
4095 GOSUB9200:GOSUB9200
4110 NEXT
4115 GOSUB9200:PRINT
4119 MW=0
4120 FORT=1T04
4130 MW=MW+(CC(T)*MD(T))
4135 NEXT
4140 S4="TOTAL DES SALAIRES DE LA SEMAINE="I00SUB9100
4145 PRINTMW:"PIECES D'OR"
4150 GOSUB9200
4155 WT=WT+MW
4160 S4="TOTAL DES SALAIRES ACTUELLEMENT ="I00SUB9100
4165 PRINTWT:"PIECES D'OR "
4170 GOSUB9200
4175 PRINT"IL VOUS RESTE "IHD1"PIECES D'OR"
4180 PRINT: S4="I00SUB9100
4190 GETI0:IFI0=""THENA190
4199 RETURN
4200 REM RAPPORT PROVISIONS
4205 PRINTCHR(147)
4206 PRINT"APRES"(MK-1)"SEMAINES":GOSUB9200
4210 S4="VOUS AVEZ="I00SUB9100

```

```

4215 S4="LES PROVISIONS SUIVANTES ="I00SUB9100
4220 PRINT:GOSUB9200
4225 FORT=1T04
4225 IFPA(T)=BORRA(T)=999THENA240
4230 PRINTPA(T)I06(T)"S DE "I06(T)
4232 X=(PA(T)/CN=PN(T))
4235 PRINT"(ABSEZ POUR "IINT(X)" SEMAINES)"
4239 GOSUB9200
4240 NEXT
4250 PRINT: S4="I00SUB9100
4255 GETI0:IFI0=""THENA295
4299 RETURN
4300 REM RAPPORT MARCHANDISES DIVERSES
4305 PRINTCHR(147)
4306 PRINT"APRES"(MK-1)"SEMAINES "I00SUB9200
4310 S4="VOUS AVEZ AUSSI:"="I00SUB9100
4320 PRINT:GOSUB9200
4322 IFOR(1)=THENA332
4325 PRINTOR(1)I06="FLACONS DE MEDICAMENTS="I00SUB9100
4330 GOSUB9200
4332 IFOR(2)=THENA342
4335 PRINTOR(2)I06="FUSILS="I00SUB9100
4340 GOSUB9200
4342 IFOR(3)=THENA352
4345 PRINTOR(3)I06="BACS DE SEL="I00SUB9100
4350 GOSUB9200
4352 IFOR(4)=THENA362
4355 PRINTOR(4)I06="BALLOTS DE TISSU="I00SUB9100
4360 GOSUB9200
4362 IFOR(5)=THENA372
4365 PRINTOR(5)I06="COUTEAUX="I00SUB9100
4370 GOSUB9200
4372 IFOR(6)=THENA380
4375 PRINTOR(6)I06="BIJOUX="I00SUB9100
4380 GOSUB9200:PRINT
4382 PRINT"IL VOUS RESTE "IHD1I06="PIECES D'OR="I0
GOSUB9100
4384 GOSUB9200
4387 PRINT: S4="I00SUB9100
4390 GETI0:IFI0=""THENA390
4399 RETURN
5000 REM TOTAL DES SALAIRES
5005 IFMT>MOTHS10
5000 DOTO5000
5010 PRINTCHR(147)
5020 PRINT:PRINT:PRINT
5025 S4="L'EQUIPAGE A CRU COMPRENDRE="I00SUB9100
5030 S4="QUE VOUS N'AVEZ PLUS ASSEZ D'ARGENT="I00SUB9100
5035 S4="POUR PAYER TOUT LE MONDE="I00SUB9100
5040 S4="EN FIN DE VOYAGE="I00SUB9100
5045 GOSUB9200:PRINT
5050 S4="ET IL N'EST PAS CONTENT !"="I00SUB9100
5055 GOSUB9200:PRINT
5060 S4="ESPERONS QUE VOUS PEREZ="I00SUB9100
5065 S4="DES PROFITS !"="I00SUB9100
5066 GOSUB9200
5070 PRINT: S4="I00SUB9100
5080 GETI0:IFI0=""THEN5080
5099 RETURN
5100 REM DISTRIBUTION RATIONS
5105 PRINTCHR(147)
5105 S4=" DISTRIBUTION DES RATIONS="I00SUB9100
5106 S4=" "I00SUB9100
5107 GOSUB9200:PRINT"SEMAINE "MK:PRINT
5108 H4="N"
5110 FORT=1T04
5112 HR(T)=1
5115 IFPA(T)>THEN5100
5120 PRINT"PLUS DE "I06(T):GOSUB9200
5130 S4="L'EQUIPAGE S'AFFAIBLIT "I00SUB9100
5135 MF=10:I00SUB9300
5139 DOTO5200
5140 X=(PN(T)+CN)*(JL-MK+1)
5165 IFPA(T)<XTHEN5200
5190 DOTO5270
5200 PRINT"VOUS MANQUEZ DE "I06(T)
5205 GOSUB9200
5210 S4="VOULEZ-VOUS METTRE L'EQUIPAGE="I00SUB9100
5215 PRINT"R DENI-RATIONS DE "I06(T)
5220 INPUTI0:I0=LEFT(I0,1)
5221 IFI0<>"O"ANDI0<>"N"THEN5220
5225 IFI0="N"THEN5270
5230 HR(T)=5I0="O"
5240 MF=5000:GOSUB9300
5250 S4="L'EQUIPAGE S'AFFAIBLIT ="I00SUB9100
5270 X=PN(T)+HR(T)=CR
5272 IFX>PA(T)THENX=PA(T)
5275 PA(T)=PA(T)-X
5280 IFPA(T)=THENPA(T)=999
5285 PRINTX:"DISTRIBUTION DE "I06(T)"S DE "I06(T)
5290 PRINT:GOSUB9200:NEXT
5295 PRINT: S4="I00SUB9100
5298 GETI0:IFI0=""THEN5298
5299 RETURN

```





Canaux de pensée

Nous poursuivons notre série sur le SE du Spectrum par l'étude des canaux à travers lesquels le micro envoie des données à l'écran et à l'imprimante ZX.

Sur le Spectrum de Sinclair, le moyen habituel d'entrée est le clavier, et les sorties sont effectuées habituellement soit sur un écran, soit sur une imprimante ZX. Dans le système Spectrum, chacun de ces matériels est appelé *canal* entrée ou sortie. L'écran, par exemple, est un canal sortie. L'ensemble des données entrant ou sortant de l'ordinateur, intervenant sous la forme de caractères à l'écran et d'informations depuis le clavier, est appelé *flux* (ou train). Un flux de données peut être dirigé dans différents canaux matériels, en supposant que le matériel du canal soit capable de traiter correctement le flux.

Nous commençons notre examen du système d'E/S Spectrum en considérant la routine « sortir un caractère », qui se trouve à l'adresse &0010. Le caractère envoyé à cette routine est dirigé vers l'écran ou l'imprimante, selon le canal préalablement sélectionné. Notre diagramme montre les différents canaux sur Spectrum et les numéros de flux qui y sont associés. Les canaux sont désignés par une seule lettre et les flux par un nombre. Sur le Spectrum non étendu, seuls les flux 0 à 3 sont actifs. Le SE Spectrum alloue le flux aux canaux comme indiqué dans le tableau en marge. Ainsi, lorsque nous voulons sortir un caractère sur

sur un appareil donné, nous devons d'abord dire au SE Spectrum quel flux nous voulons avoir. Pour écrire sur l'écran, nous sélectionnons le flux 2, puisque c'est lui qui est associé au canal S, et pour cela nous utilisons une routine ROM à l'adresse &1601 pour dire au SE quel flux nous voulons sélectionner. Le numéro de flux est placé dans le registre A, avant d'appeler la routine, qui ouvre alors le canal matériel associé à ce numéro de flux. Par exemple, pour ouvrir le canal S pour la sortie, nous exécutons les instructions suivantes :

```
LD  A,2
CALL &1601
```

Une fois qu'un canal est ouvert, il suffit, pour y envoyer un caractère, de mettre le code du caractère dans le registre A, puis d'exécuter une instruction RTS pour appeler la routine à l'adresse &0010.

Le canal S opère sur la zone de l'écran qui est accessible via les instructions BASIC normales PRINT. Toutefois, il est aussi possible d'accéder aux deux lignes inférieures de l'écran qui sont normalement utilisées par l'interpréteur BASIC pour la sortie de messages d'erreurs et sugges-

Flux	Canal
0	K
1	K
2	S
3	P





tions. Comme le montre la figure, ces lignes font partie du canal K.

Pour envoyer un caractère vers ces lignes, nous ouvrons simplement le canal K à l'aide de :

```
LD A,0
CALL &1601
```

De même, le canal imprimante peut être ouvert en chargeant le registre A avec 3.

Comme les caractères standard, nous pouvons envoyer des caractères de contrôle vers n'importe quel canal. Évidemment, l'effet qu'auront ces caractères de contrôle dépend du canal utilisé, mais cela n'implique pas que nous puissions avoir les équivalents en langage machine de PRINT AT, PRINT INK, PRINT PAPER, etc. Le tableau suivant montre quelques codes de contrôle utiles et leurs fonctions lorsqu'ils passent sur le canal S ou K. Bien sûr, certains n'auront pas d'effet s'ils sont envoyés vers l'imprimante ZX (via canal P).

Code	Paramètres	Effet
8	—	Curseur à gauche d'un espace.
9	—	Curseur à droite d'un espace.
10	—	Curseur vers le bas d'une ligne.
11	—	Curseur vers le haut d'une ligne.
12	—	Effacer.
13	—	ENTER.
16	n	INK n (nécessite un octet de plus).
17	n	PAPER n (nécessite un octet de plus).
18	n	FLASH n (nécessite un octet de plus).
19	n	BRIGHT n (nécessite un octet de plus).
20	n	INVERSE n (nécessite un octet de plus).
21	n	OVER n (nécessite un octet de plus).
22	nn	AT vx (nécessite deux octets de plus).
23	n	TAB n (nécessite un paramètre de plus).

Les octets supplémentaires requis par certains codes de contrôle sont les paramètres qui les suivraient normalement dans une instruction BASIC. Par exemple, pour exécuter une commande PAPER 3, nous enverrions simplement les octets 17 et 3 vers le canal S. La partie suivante d'un programme montre l'équivalent en langage machine de PRINT AT 10,10; « A ».

```
3E02 10 ld a,2 open channel S by
CD0116 20 call #1601 selecting stream 2 -
3E16 30 ld a,22 token for AT
D7 40 rst #10 output a char routine
3E0A 50 ld a,10 by co-ordinate
D7 60 rst #10
3E0A 70 ld a,10 by co-ordinate
D7 80 rst #10
3E41 90 ld a,65 ASCII code for 'A'
D7 100 rst #10
C9 110 ret
```

Pas trop difficile, n'est-ce pas? Une exception notable dans la liste des codes de contrôle est l'absence de code pour CLS. Pour effacer l'écran, nous devons appeler une autre routine ROM à l'adresse &06DB. Il est essentiel d'ouvrir le canal S avant de l'appeler, et il est également nécessaire de rouvrir le canal S après l'avoir utilisé, si nous voulons afficher autre chose à l'écran. Cette routine efface l'écran :

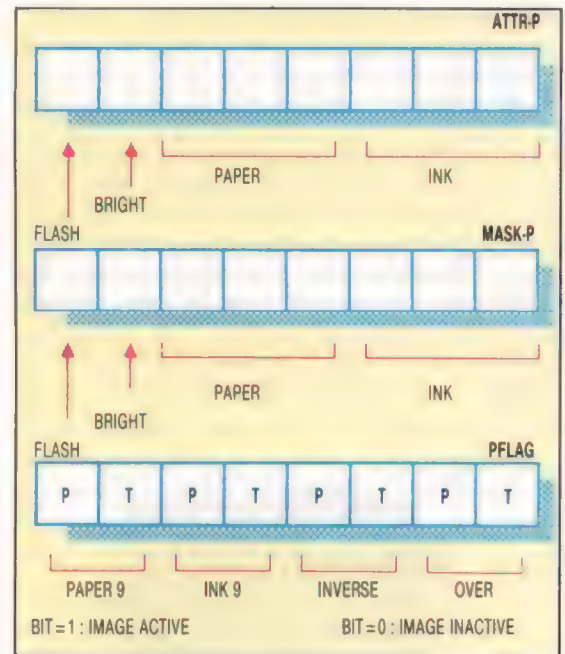
```
3E02 10 ld a,2
CD0116 20 call #1601 open channel S
CDB806 30 call #06DB clear the screen
3E02 40 ld a,2
CD0116 50 call #1601 re-open channel S
C9 60 ret
```

Un attribut utile de la routine « sortir un caractère » sur un Spectrum standard est que les nombres ayant été passés, qui représentent des mots clés BASIC, sont affichés en entier. Ainsi :

```
LD 1,249
RTS &10
```

vont — en supposant que les canaux S, P ou K ont été sélectionnés — afficher le mot clé RANDOMIZE à l'écran.

Les commandes graphiques, telles que PAPER, INK et BRIGHT, émises par la routine « sortir un caractère », ne sont effectives que sur cette séquence de caractères de sortie — on dit que ce sont des *articles couleur temporaire*. La commande PAPER n, si elle est émise par une instruction PRINT, est une commande *permanente*, et elle reste effective jusqu'à ce qu'une autre commande PAPER soit émise.



Mike Clowes

Les deux variables système intéressantes sont appelées ATTR-P et MASK-P. La figure ci-dessus montre comment ces variables système contrôlent différents aspects de l'affichage. Dans ATTR-P, les trois bits qui contrôlent les couleurs PAPER et INK sur une base permanente se voient donner les valeurs indiquées en haut de la colonne suivante. Si le bit pour FLASH ou BRIGHT est mis à 1, alors le dispositif est opérationnel. ATTR-P est à l'adresse 23693.

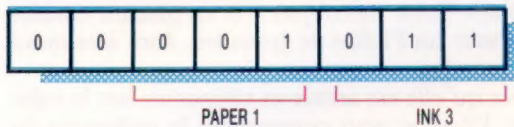
MASK-P est à l'adresse 23694, et tout bit mis à 1 dans cet octet assure que les attributs sur l'écran à la position voulue ne sont pas altérés par le contenu de ATTR-P. Une autre variable système utile est appelée PFLAG, et elle se trouve à l'adresse 23697. Cette variable figure aussi sur le diagramme. Elle est utilisée par le SE pour indiquer PAPER 9, INK 9, INVERSE et OVER.



Bits	Couleur
000	Noir
001	Bleu
010	Rouge
011	Magenta
100	Vert
101	Cyan
110	Jaune
111	Blanc

Il y a deux autres variables système — ATTR-T et MASK-T — aux adresses 23695 et 23696 respectivement. Elles sont disposées de la même manière que ATTR-P et MASK-P, mais contrôlent les couleurs temporaires (celles qui servent dans les instructions PRINT ou mises par des codes de contrôle par la routine RTS &10).

Il est très facile d'utiliser ATTR-P et MASK-P pour mettre des couleurs; il suffit de manipuler les contenus des variables système, en ne modifiant que les bits nécessaires. C'est facile à faire à l'aide des instructions logiques du Z80, AND et OR. Ainsi, pour exécuter une commande PAPER 1 INK 3, nous devons mettre ATTR-P comme suit :



à l'aide de cette partie de code :

```
LD    A,11
LD    (23693),A
RET
```

Naturellement, comme avec les commandes BASIC PAPER et INK, les nouvelles couleurs n'affecteront pas ce qui a été écrit auparavant, et l'écran ne passera pas à la nouvelle couleur PAPER jusqu'à ce qu'une commande CLS, ou son équivalent, soit exécutée.

Les routines graphiques sur le Spectrum font partie du programme interpréteur du BASIC. Considérons à présent les opérations PLOT et DRAW. En utilisant ces routines, il est facile d'apporter des changements en altérant les variables ATTR-P et MASK-P, comme nous l'avons déjà écrit.

Les routines elles-mêmes sont faciles à utiliser. La première, PLOT, est appelée à l'adresse &22E5. Ses coordonnées sont transmises dans la paire de registres BC (B contenant la coordonnée y et C la coordonnée x). Ainsi, pour exécuter la commande PLOT 100,100 à partir d'un programme en langage machine, il suffit d'exécuter ces instructions :

```
LD    B,100    ;coordonnée y
LD    C,100    ;coordonnée x
CALL  &22E5    ;le faire
RET
```

Les changements de couleur sont faciles à effectuer; la routine suivante trace un point rouge sur l'écran, puis restaure ATTR-P à son état précédent avant de retourner au BASIC.

```
3A05C 10    ld a,(23693)    test ATTR-P in A res
F5     20    push af         f and save on stack
E6FB  30    and 248        fclear low 3 bits to 0
F602  40    or 2         fset bit 1 to give red ink
0664  50    ld b,100     f y co-ordinate
0E64  60    ld c,100     f x co-ordinate
CDE522 70    call &22e5     fcall PLOT
F1     80    pop af         f restore original
3205C 90    ld (23693),a    fATTR-P contents
C9     100   ret
```

Jusqu'à présent, nous nous sommes limités au canal S. Qu'en est-il du canal K, qui a aussi des possibilités de sortie? Nous pouvons écrire sur la partie inférieure de l'écran si nous voulons, mais ce que nous y écrivons sera « écrasé » lorsque le SE ou l'interpréteur générera un message. Pour des entrées utilisant ce canal, nous n'avons même pas besoin d'appeler une routine ROM! Le clavier est exploré une fois toutes les 20 micro-secondes, et certaines variables système sont affectées, selon qu'une touche est appuyée ou non.

Deux des usages particuliers sont LAST-K (à l'adresse 23560), qui contient le code caractère de la dernière touche appuyée, et une variable système en 23556, qui contient la valeur 255 si une touche n'a pas été appuyée à cet instant. La routine, ici, peut donc servir à attendre qu'une touche soit appuyée, puis donner le code caractère dans le registre A.

Il vérifie simplement le contenu de l'adresse 23556 jusqu'à ce qu'il ne soit pas égal à 255. La valeur dans LAST-K sera alors celle de la touche appuyée.

```
10 GET-CHAR routine
3A045C 20 key: ld a,(23556) fcheck to see if
FEFF  30    cp 255        fkey is being pressed
2BF9  40    jr .z,key     fkeep checking
3A085C 50    ld a,(23560) fset LAST-K in A
C9     60    ret
```

Avant de quitter le clavier, considérons une paire de variables système qui peuvent être utiles.

Variable	Adresse	Description
REPDEL	23561	Délai en 50 ^e de seconde avant que le clavier commence à répéter. Vous pouvez modifier cette valeur pour changer le délai avant que le clavier commence à répéter.
REPPER	23562	Délai entre seconde et la touche suivante répète. Peut aussi être POKÉ.
PIP	23609	Longueur de clic généré lorsque les touches sont appuyées. Les valeurs suivantes donnent un bip.

L'autre canal utilisé sur un Spectrum non étendu est le canal P, qui sert à l'imprimante ZX. Il est généralement associé au flux 3. Ainsi :

```
LD    A,3
CALL  &1601
```

ouvrira le canal P et les caractères suivants apparaîtront sur l'imprimante ZX si elle est connectée. Il est évident que BASIC fait grand usage des différents canaux : PRINT et LIST utilisent le canal S, LPRINT et LLIST, le canal P, et INPUT, le canal K. On peut encore y ajouter nos propres canaux, afin de pouvoir aisément accéder à des appareils supplémentaires tels que Microdrives, imprimantes différentes et autres matériels.



Iconoclaste

Le scénario de ce jeu est sans doute familier, mais la commande du curseur par sélection d'icônes ajoute une dimension intéressante au jeu Shadowfire de Beyond Software.

Les acteurs du drame

Nous présentons ici trois personnages commandés par le joueur dans diverses situations de Shadowfire. A gauche, Torik apparaît en mode déplacement, les diverses directions possibles étant mises en valeur. Au centre, Zark Mondor est attaqué et est en mode combat. De nouveau, les directions pouvant être prises sont mises en valeur; ses adversaires apparaissent du côté gauche de l'écran. Maul, le robot de combat, apparaît avec des objets. La partie centrale, en bas de l'écran, montre les objets qu'il transporte et, à gauche, on aperçoit les objets situés près de lui, qu'il peut ramasser. Les instructions sont envoyées à l'ordinateur en plaçant le curseur (une croix blanche) sur l'icône sélectionnée et en appuyant sur le bouton de mise à feu du manche à balai.

Les jeux sur ordinateur deviennent de plus en plus perfectionnés. Au lieu de se borner à reproduire les jeux de café ou les jeux d'aventures, les concepteurs développent de plus en plus un style qui leur est propre. Ces jeux intègrent de nombreuses fonctions des jeux de café et de stratégies, afin de produire un divertissement qui dure plus longtemps que les cinq minutes traditionnelles et sont encore plus stimulants que les énigmes proposées par les jeux d'aventures.

Le scénario de Shadowfire est le suivant : le général Zoffa capture l'ambassadeur Kryxix qui détient les plans d'un nouveau type de vaisseau spatial nommé Shadowfire. Vous êtes chargé, en tant que joueur, de libérer l'ambassadeur avant qu'il ne soit forcé à livrer les plans; vous avez cent minutes pour le faire. Pour vous venir en aide, vous disposez de six personnages, ayant chacun ses forces et ses faiblesses.

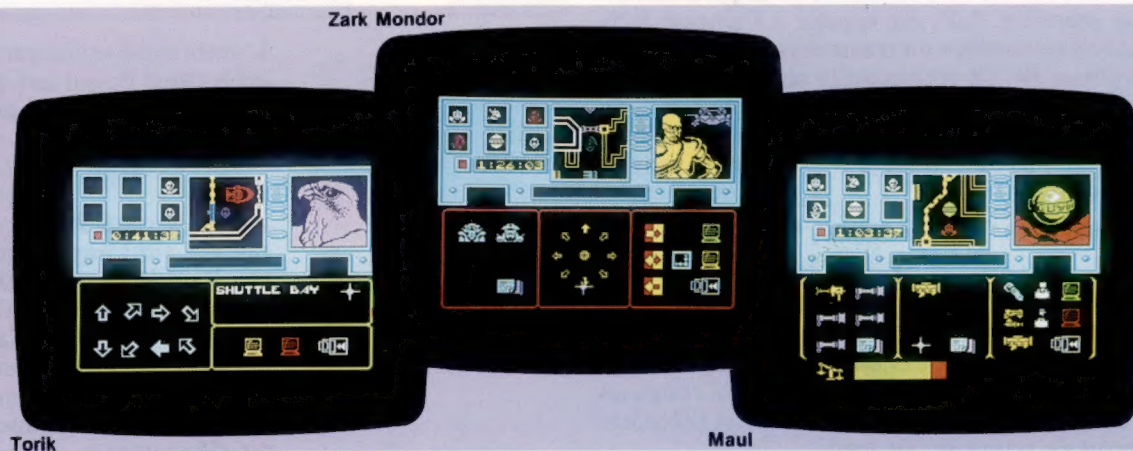
Pour monter sur le vaisseau spatial de Zoffa, votre équipe doit y être « téléportée ». Le seul membre qui puisse organiser cette intervention est Manto; il doit donc être d'abord envoyé pour poser le faisceau de « téléportation », afin de permettre aux autres de le suivre. Cependant, avant de les lancer dans cette aventure, il est préférable d'équiper chacun d'eux en fonction de ses forces et de ses faiblesses avec les armes disponibles.

Shadowfire utilise un système unique de déplacement des personnages, qui leur permet de saisir des objets et de livrer des combats. Contrairement aux autres types d'aventures où l'utilisateur doit entrer des commandes comme « Aller au nord », « Saisir laser », il est possible d'exécuter avec ce jeu toutes les fonctions au moyen d'icônes et de déplacements du curseur — un peu comme le système d'exploitation du Macintosh

d'Apple. Pour munir le chef de l'équipe, Zark, de grenades par exemple, vous sélectionnez d'abord l'icône Zark. Lorsqu'il est choisi, l'écran propose une image montrant sa force, sa résistance, et d'autres attributs. Sur le côté droit de l'écran, trois icônes « moniteur » représentent le mouvement, le mode de combat et l'écran des objets.

En sélectionnant les objets avec le curseur (qui peut être déplacé au moyen du clavier, d'un manche à balai, ou d'un crayon optique), l'écran change de nouveau pour afficher les objets qui se trouvent près du personnage ainsi qu'un certain nombre d'icônes « activité ». En choisissant l'icône « Saisir des objets » et en plaçant ensuite le curseur sur l'icône de grenades, Zark sera muni de grenades.

Dès qu'elle est armée et téléportée sur le vaisseau, l'équipe peut commencer la recherche du général Zoffa et de l'ambassadeur Kryxix. Le vaisseau comprend de nombreuses pièces et corridors, où l'on trouve des armes et des clés permettant d'ouvrir les portes, et des gardes armés que vous devez détruire avant de pouvoir poursuivre. Si vous ne trouvez pas la clé appropriée, vous pouvez faire appel aux services de Sevrina qui forcera les verrous. Pour exploiter au maximum les cent minutes dont vous disposez, vous devez mettre au point une stratégie. Certains personnages sont plus efficaces que d'autres dans certaines situations. Il est préférable de veiller à utiliser le bon personnage, à l'endroit approprié, au bon moment. C'est surtout le mode de commande du curseur qui rend ce jeu intéressant. Avec les icônes, le joueur est en mesure de réagir beaucoup plus rapidement à des situations que s'il avait à entrer chaque commande séparément.



Mike Clowes

**Page manquante
(publicité)**

**Page manquante
(publicité)**